A poor man's epistemic logic based on propositional assignment and higher-order observation

Andreas Herzig, Emiliano Lorini, and Faustine Maffre

University of Toulouse, IRIT, 118, Route de Narbonne, F-31062 Toulouse, France

Abstract. We introduce a dynamic epistemic logic that is based on what an agent can observe, including joint observation and observation of what other agents observe. This generalizes van der Hoek, Wooldridge and colleague's logics ECL-PC(PO) and LRC where it is common knowledge which propositional variables each agent observes. In our logic, facts of the world and their observability can both be modified by assignment programs. We show how epistemic operators can be interpreted in this framework and identify the conditions under which the principles of positive and negative introspection are valid. We also provide a sound and complete axiomatization and prove that the satisfiability problem is PSPACE-complete. Finally, we show how public and private announcements can be expressed and illustrate the latter by the gossip spreading problem.

1 Introduction

In recent years, several authors investigated how an epistemic logic could be grounded on the notion of visibility (or observability) of propositional variables, most prominently Epistemic Coalition Logic of Propositional Control with Partial Observability ECL-PC(PO) [14] and Logic of Revelation and Concealment LRC [13]. The idea is that each agent has a set of propositional variables she can observe: no different truth value is possible for her. The other way round, any combination of truth values of the non-observable variables is possible for her.

A disadvantage of ECL-PC(PO) and LRC is that what each agent can see is common knowledge. This is a strong hypothesis that we are going to relax in the present paper. While in ECL-PC(PO) and LRC, visibility information is in terms of propositional variables associated to agents, we here consider propositional variables associated to *sequences* of agents. Syntactically, we represent this by means of atomic formulas that we call *visibility atoms*. They take the form $S_{i_1}\, S_{i_2}...S_{i_n}\, p$, where p is a propositional variable and $i_1, i_2, ..., i_n$ are agents. When $n{=}0$ then we have nothing but a propositional variable. For $n{=}1$, the atom $S_{i_1}\, p$ reads "agent i_1 sees the value of the variable p", and for $n{=}2$, the second-order observation $S_{i_1}\, S_{i_2}\, p$ reads "agent i_1 sees whether i_2 sees the value of p"; and so on.

Our models are simply sets of visibility atoms. In order to guarantee positive and negative introspection we have to ensure that agents are always aware of what they see: for every agent i and propositional variable p, we require $S_i\, S_i\, p$ to be in every valuation. More generally, we say that a valuation V is *introspective* when it contains every visibility atom having two consecutive S_i, such as $S_j\, S_i\, S_i\, S_k\, p$.

Visibility information allows to interpret epistemic operators: for propositional variables p, the formula $K_i p$ is true in a valuation V if V contains both p and $S_i\, p$. More generally, the truth condition for $K_i\varphi$ is based on a relation between valuations that can be defined from our visibility atoms: $V \sim_i V'$ if every atom that i sees in V has the same truth value in V and in V'. While the relations $\sim_i$ are reflexive everywhere, they are symmetric and transitive—and therefore equivalence relations—on the set of introspective valuations only. The truth condition for the epistemic operator then takes the standard form: $K_i\varphi$ is true in V if φ is true in every valuation related to V by $\sim_i$. The positive and negative introspection axioms $K_i\varphi \to K_iK_i\varphi$ and $\neg K_i\varphi \to K_i\neg K_i\varphi$ are valid in the set of introspective valuations.

A further novelty of our approach as compared to existing visibility-based epistemic logics is that we also account for common knowledge: our language includes a special atomic formula for joint attention of the form JSp that reads "all agents jointly see the value of p". Metaphorically, joint attention about a propositional variable p can be understood as eye contact between the agents when observing p. Just as individual visibility, we generalize our account to higher-order visibility, adding a constraint on valuations that guarantees introspection of common knowledge. We moreover require that joint visibility implies individual visibility by imposing that $S_i\, p \in V$ whenever $JS\, p \in V$. We can then interpret a modal operator of common knowledge CK in the same way as the modal operator of individual knowledge.

Just as several existing proposals, we take inspiration from dynamic epistemic logics DEL [5] and add dynamics to our observation-based epistemic logic. Specifically, we adapt van der Hoek et al.'s logic LRC which has two update operations modifying visibility: revealing and concealing the value of a variable to some agent. These two primitives can however not be taken over as they stand because the naive update of a valuation may no longer be introspective. We exclude this by an appropriate definition of update. We relate our assignment programs to Dynamic Logic of Propositional Assignments DL-PA [12,4], which is a dialect of Propositional Dynamic Logic PDL [8] where PDL's abstract atomic programs are instantiated by assignments of truth values to atomic formulas. The benefit of that link is a PSPACE upper bound of the complexity of both satisfiability and model checking. Moreover, visibility updates can capture public and private announcements of visibility atoms and negations thereof.

We call our logic DEL-PAO: Dynamic Epistemic Logic of Propositional Assignment and Observation. The paper is organized as follows: sections 2 and 3 introduce language and semantics of DEL-PAO. Sections 4 and 5 contain an axiomatization and the complexity result. Section 6 illustrates our logic by two applications: the embedding of announcements and a modeling of the gossip spreading problem. Section 7 discusses related work and Section 8 concludes.[1]

2 Language

Let *Prop* be a countable non-empty set of propositional variables and let *Agt* be a finite non-empty set of agents. Atomic formulas of our language are sequences of visibility operators followed by propositional variables. The formal definition is as follows.

[1] A version of this paper including proofs and a further case study (the coordinated attack problem) is available at `http://www.irit.fr/~Andreas.Herzig/P/Lori15.html`.

The set of *observability operators* is

$$OBS = \{S_i : i \in Agt\} \cup \{JS\},$$

where S_i stands for individual visibility of agent i and JS stands for joint visibility of all agents. The set of all sequences of visibility operators is noted OBS^* and the set of all non-empty sequences is noted OBS^+. We use $\sigma, \sigma', \ldots$ for elements of OBS^*. Finally, the set of atomic formulas is

$$ATM = \{\sigma\, p : \sigma \in OBS^*, p \in Prop\}.$$

The elements of that set are also called *visibility atoms*, or atoms for short. For example, $JS\, S_2\, q$ reads "all agents jointly see whether agent 2 sees the value of q"; in other words, there is joint attention in the group of all agents concerning 2's observation of q. We use $\alpha, \alpha', \ldots, \beta, \beta', \ldots$ for elements of ATM.

The language of DEL-PAO is then defined by the following grammar:

$$\pi ::= +\alpha \mid -\alpha \mid \pi;\pi \mid \pi \sqcup \pi \mid \varphi?$$
$$\varphi ::= \alpha \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid CK\varphi \mid [\pi]\varphi$$

where α ranges over ATM and i over Agt.

Our atomic programs are assignments of truth values to atoms from ATM: $+\alpha$ makes α true and $-\alpha$ makes α false. Complex programs are constructed with dynamic logic operators: $\pi;\pi'$ is sequential composition, $\pi \sqcup \pi'$ is nondeterministic choice, and $\varphi?$ is test. Just as in dynamic logic, the formula $[\pi]\varphi$ reads "after every execution of π, φ is true". The formula $K_i\varphi$ reads "i knows that φ is true on the basis of what she observes", and $CK\varphi$ reads "all agents jointly know that φ is true on the basis of what they jointly observe". Our epistemic operators account for forms of individual and common knowledge that are respectively obtained via individual observation and joint observation of facts. This differs therefore conceptually from the classical operators of individual and common knowledge as studied in the area of epistemic logic [6]. We will come back to this in Section 3.4.

The other boolean operators $\top$, $\bot$, $\vee$, $\rightarrow$ and $\leftrightarrow$ are defined as usual, and $\widehat{K}_i\varphi$ abbreviates $\neg K_i \neg\varphi$. The program *skip* abbreviates $\top?$ and *fail* abbreviates $\bot?$. We also use the abbreviation π^k, for $k \geq 0$, inductively defined by $\pi^0 = skip$ and $\pi^{k+1} = \pi^k;\pi$.

The set of atomic formulas of ATM occurring in the formula φ is noted $ATM(\varphi)$; the set $ATM(\pi)$ is defined similarly. For example, $ATM(q?;+S_2\, p) = \{q, S_2\, p\}$ and $ATM([\pi]S_1\, JS\, p \rightarrow q) = \{q, S_2\, p, S_1\, JS\, p\}$. (So $JS\, p$ is not an atom of the latter.) The length of formulas φ and programs π, noted $length(\varphi)$ and $length(\pi)$, is the number of symbols used to write them down, where we do not count [,] and parentheses and consider that the length of JS, CK, agent names and propositional variables is 1. For example, $length(S_2\, S_2\, p) = 5$ and $length([+S_2\, p]JS\, p \wedge q) = 8$.

3 Semantics

We define valuations and stipulate constraints that are motivated by the requirement that visibility information should be introspective and that joint visibility should imply individual visibility. We then define indistinguishability relations between valuations and interpret formulas and programs.

3.1 Introspective valuations

A *valuation* is a subset of the set of atoms ATM. A valuation $V \in 2^{ATM}$ is *introspective* if and only if the following hold, for every $\alpha \in ATM$ and $i \in Agt$:

$$S_i\, S_i\, \alpha \in V \tag{C1}$$
$$JS\; JS\, \alpha \in V \tag{C2}$$
$$JS\, S_i\, S_i\, \alpha \in V \tag{C3}$$
$$\text{if } JS\, \alpha \in V\text{, then } S_i\, \alpha \in V \tag{C4}$$
$$\text{if } JS\, \alpha \in V\text{, then } JS\, S_i\, \alpha \in V \tag{C5}$$

The set of all introspective valuations is noted $INTR$.

(C1) is about introspection of individual sight: an agent always sees whether she sees the value of an atom. (C2) requires the same for joint sight; indeed, if $JS\, \alpha$ is true then $JS\; JS\, \alpha$ should be true by introspection, and if $JS\, \alpha$ is false then all agents jointly see that at least one of them has broken eye contact. (C3) forces the first to be common knowledge. (C4) guarantees that joint visibility implies individual visibility. Together with (C2), (C5) guarantees that $JS\, \alpha \in V$ implies $JS\, \sigma\, \alpha \in V$ for $\sigma \in OBS^*$.[2]

The constraints C4 and C5 ensure that $JS\, \alpha \in V$ implies $\sigma\, \alpha \in V$ for $\sigma \in OBS^+$. This motivates the following relation of *introspective consequence* between atoms: $\alpha \rightsquigarrow \beta$ iff either $\alpha = \beta$, or $\alpha = JS\, \alpha'$ and $\beta = \sigma\, \alpha'$ for some $\sigma \in OBS^+$.

Closure under introspective consequence characterizes introspective valuations.

Proposition 1. *A valuation $V \subseteq ATM$ is introspective if and only if, for every $\alpha, \beta \in ATM$ and $i \in Agt$:*

$$\sigma\, S_i\, S_i\, \alpha \in V \textit{ for every } \sigma \in OBS^* \tag{1}$$
$$\sigma\, JS\, \alpha \in V \textit{ for every } \sigma \in OBS^+ \tag{2}$$
$$\textit{if } \alpha \in V \textit{ and } \alpha \rightsquigarrow \beta \textit{ then } \beta \in V \tag{3}$$

Call an atom $\alpha \in ATM$ is *valid in $INTR$* if and only if α belongs to every valuation in $INTR$. By Proposition 1, α is valid in $INTR$ if and only if α is of the form either $\sigma\, S_i\, S_i\, \alpha$ with $\sigma \in OBS^*$, or $\sigma\, JS\, \alpha$ with $\sigma \in OBS^+$.

Observe that we do not impose the constraint "if $\sigma\, \alpha \in V$ for every $\sigma \in OBS^*$ then $JS\, \alpha \in V$", which corresponds to the greatest fixed point definition of the operator of common knowledge from shared knowledge. We will comment on this in Section 3.4.

3.2 Indistinguishability relations

Two valuations are related by the indistinguishability relation for agent i, noted $\sim_i$, if every α that i sees has the same value. Similarly, we have a relation $\sim_{Agt}$ for joint indistinguishability. They are defined as follows:

$$V \sim_i V' \text{ iff } S_i\, \alpha \in V \text{ implies } V(\alpha) = V'(\alpha)$$
$$V \sim_{Agt} V' \text{ iff } JS\, \alpha \in V \text{ implies } V(\alpha) = V'(\alpha)$$

where we write $V(\alpha) = V'(\alpha)$ when either $\alpha \in V$ and $\alpha \in V'$, or $\alpha \notin V$ and $\alpha \notin V'$.

[2] We need (C2) when σ contains JS: in order to prove that $JS\, \alpha \in V$ implies $JS\, S_i\, JS\, \alpha \in V$ we use that $JS\; JS\, \alpha \in V$ by (C2) and that $JS\; JS\, \alpha \in V$ implies $JS\, S_i\, JS\, \alpha \in V$ by (C5).

The binary relations $\sim_i$ and $\sim_{Agt}$ are reflexive. They are neither transitive nor symmetric: for example, $\emptyset \sim_i V$ for every $V \subseteq ATM$, while $V \not\sim_i \emptyset$ as soon as there is a p such that p and S_i are in V. However, both properties hold on valuations satisfying the introspection constraints (C1) and (C2).

Proposition 2. *The relation $\sim_{Agt}$ and every $\sim_i$ are equivalence relations on $INTR$.*

Lemma 1. *Let $V \in INTR$, $V' \in 2^{ATM}$. If $V \sim_i V'$ or $V \sim_{Agt} V'$ then $V' \in INTR$.*

3.3 Truth conditions and validity

Given an introspective valuation V, our update operations add or remove atoms from V. This requires some care: we want the resulting valuation to be introspective. For example, removing $S_i\, S_i\, p$ should be impossible. Another example is when V does not contain $S_i\, p$: then $V \cup \{JS\, p\}$ would violate (C4). So when adding an atom to V we also have to add all its *positive consequences*. Symmetrically, when removing an atom we also have to remove its *negative consequences*. Let us define the following:

$$Eff^+(\alpha) = \{\beta \in ATM : \alpha \rightsquigarrow \beta\}$$
$$Eff^-(\alpha) = \{\beta \in ATM : \beta \rightsquigarrow \alpha\}$$

Clearly, when V is introspective then both $V \cup Eff^+(\alpha)$ and $V \setminus Eff^-(\alpha)$ are so, too (unless α is valid).

Now the truth conditions are as follows:

$$\begin{array}{ll} V \models \alpha & \text{iff } \alpha \in V \\ V \models \neg\varphi & \text{iff } V \not\models \varphi \\ V \models \varphi \wedge \psi & \text{iff } V \models \varphi \text{ and } V \models \psi \\ V \models K_i\varphi & \text{iff } V' \models \varphi \text{ for all } V' \text{ such that } V \sim_i V' \\ V \models CK\varphi & \text{iff } V' \models \varphi \text{ for all } V' \text{ such that } V \sim_{Agt} V' \\ V \models [\pi]\varphi & \text{iff } V' \models \varphi \text{ for all } V' \text{ such that } VR_\pi V' \end{array}$$

where R_π is a binary relation on valuations that is defined (by mutual recursion with the definition of $\models$) by:

$$\begin{array}{ll} VR_{+\alpha}V' & \text{iff } V' = V \cup Eff^+(\alpha) \\ VR_{-\alpha}V' & \text{iff } V' = V \setminus Eff^-(\alpha) \text{ and } \alpha \text{ is not valid in } INTR \\ VR_{\pi_1;\pi_2}V' & \text{iff there is } U \text{ such that } VR_{\pi_1}U \text{ and } UR_{\pi_2}V' \\ VR_{\pi_1 \sqcup \pi_2}V' & \text{iff } VR_{\pi_1}V' \text{ or } VR_{\pi_2}V' \\ VR_{\varphi?}V' & \text{iff } V = V' \text{ and } V \models \varphi \end{array}$$

The relation R_π is defined just as in PDL for the program operators ;, $\sqcup$ and ?. The interpretation of assignments is designed in a way such that we stay in $INTR$: the program $+\alpha$ adds all the positive consequences of α; the program $-\alpha$ fails if α is valid in $INTR$ and otherwise removes all the negative consequences of α. For example, we never have $VR_{-S_1\, S_1\, p}V'$, i.e., the program $-S_1\, S_1\, p$ always fails. In contrast, the program $-S_1\, S_2\, p$ always succeeds, and we have $VR_{-S_1\, S_2\, p}\, (V \setminus \{S_1\, S_2\, p, JS\, S_2\, p, JS\, p\})$ because the only atoms—beyond $S_1\, S_2\, p$ itself—whose consequence is $S_1\, S_2\, p$ are $JS\, S_2\, p$ and $JS\, p$. Therefore $V \not\models [-S_1\, S_2\, p]JS\, p$ for every V. We say that two programs π_1 and π_2 are equivalent, noted $\pi_1 \equiv \pi_2$, if and only if R_{π_1} equals R_{π_2} on

$INTR$, i.e., for every $V, V' \in INTR$ we have $V R_{\pi_1} V'$ iff $V R_{\pi_2} V'$. For example, $-S_i\, S_i\, p \equiv \mathit{fail}$.

Lemma 2. *Let $V \in INTR$ and $V R_\pi V'$. Then $V' \in INTR$.*

Proposition 3. *For every $V \in INTR$, $i \in Agt$ and program π, V is only related to valuations in $INTR$ by $\sim_i$, $\sim_{Agt}$ and R_π.*

When $V \models \varphi$ we say that V is a *model* of φ. The set of (not necessarily introspective) models of φ is noted $\|\varphi\|$. A formula φ is *satisfiable in $INTR$* if φ has an introspective model, i.e., if $\|\varphi\| \cap INTR \neq \emptyset$. For example, $JS\, p \wedge \neg S_i\, p$ has a model, but does not have an introspective model and is therefore unsatisfiable in $INTR$. A formula φ is *valid in $INTR$* if $INTR \subseteq \|\varphi\|$. We also say that φ is a *validity of* DEL-PAO. For example, $\neg[-S_1\, S_2\, p] JS\, p$ is valid in $INTR$. Note that $\neg\beta \to [+\alpha]\neg\beta$ is valid in $INTR$ if and only if $\alpha \not\rightsquigarrow \beta$.

Formulas without epistemic operators only depend on the atoms occurring in it.

Proposition 4. *Let φ be without epistemic operators. Let $V, V' \in 2^{ATM}$ such that $V(\alpha) = V'(\alpha)$ for every atom $\alpha \in ATM(\varphi)$. Then $V \models \varphi$ if and only if $V' \models \varphi$.*

This proposition will be instrumental in the rest of the paper. Observe that it does not hold when φ contains epistemic operators. For example, the truth value of $K_i p$ depends on that of $S_i\, p$, which however does not occur in $ATM(K_i p)$.

3.4 Discussion

Both the operators of individual knowledge and the operator of common knowledge of DEL-PAO satisfy all the principles of the standard epistemic logic S5. There are also some further validities of DEL-PAO, for example the S5-invalid formula $K_i(p \vee q) \to (K_i p \vee K_i q)$; cf. the axiom $Red_{K,\vee}$ below. This is a strong principle: to give an example, if one knows that the butler or the gardener was the murderer then one knows which of them it was. It is however shared by all visibility-based epistemic logics.

Our common knowledge operator obeys the fixed point axiom:

$$CKp \to p \wedge \Big(\bigwedge_{i \in Agt} K_i\, CKp \Big).$$

This is ensured by the fact that by constraints (C2) and (C4), the formula $\bigwedge_{i \in Agt} S_i\; JS\, p$ is valid in $INTR$. Our notion of common knowledge is however weaker than standard common knowledge because the induction axiom

$$\Big(\varphi \wedge CK \Big(\varphi \to \bigwedge_{i \in Agt} K_i \varphi \Big) \Big) \to CK\varphi$$

is invalid in $INTR$. Beyond the technical reason for that choice (such an infinitary constraint cannot be captured by formula built from visibility atoms) we follow [15,11] and assume that such a principle is too strong for a logic of common knowledge.

4 Axiomatization

The axiomatization of DEL-PAO is given by:

– the axioms of CPL (Classical Propositional Logic);

– the reduction axioms for epistemic operators:

$$K_i\alpha \leftrightarrow S_i\,\alpha \wedge \alpha \qquad (Red_{K,\alpha})$$

$$CK\alpha \leftrightarrow JS\,\alpha \wedge \alpha \qquad (Red_{CK,\alpha})$$

$$K_i\neg\alpha \leftrightarrow S_i\,\alpha \wedge \neg\alpha \qquad (Red_{K,\neg})$$

$$CK\neg\alpha \leftrightarrow JS\,\alpha \wedge \neg\alpha \qquad (Red_{CK,\neg})$$

$$K_i(\varphi \wedge \varphi') \leftrightarrow K_i\varphi \wedge K_i\varphi' \qquad (Red_{K,\wedge})$$

$$CK(\varphi \wedge \varphi') \leftrightarrow CK\varphi \wedge CK\varphi' \qquad (Red_{CK,\wedge})$$

$$K_i\left(\bigvee_{\alpha\in A^+}\alpha \vee \bigvee_{\alpha\in A^-}\neg\alpha\right) \leftrightarrow \left(\bigvee_{\alpha\in A^+}K_i\alpha\right) \vee \left(\bigvee_{\alpha\in A^-}K_i\neg\alpha\right) \qquad (Red_{K,\vee})$$

$$CK\left(\bigvee_{\alpha\in A^+}\alpha \vee \bigvee_{\alpha\in A^-}\neg\alpha\right) \leftrightarrow \left(\bigvee_{\alpha\in A^+}CK\alpha\right) \vee \left(\bigvee_{\alpha\in A^-}CK\neg\alpha\right) \qquad (Red_{CK,\vee})$$

– the reduction axioms for dynamic operators:

$$[\pi;\pi']\varphi \leftrightarrow [\pi][\pi']\varphi \qquad (Red_{;})$$

$$[\pi \sqcup \pi']\varphi \leftrightarrow [\pi]\varphi \wedge [\pi']\varphi \qquad (Red_{\sqcup})$$

$$[\varphi?]\varphi' \leftrightarrow \varphi \to \varphi' \qquad (Red_{?})$$

$$[+\alpha]\neg\varphi \leftrightarrow \neg[+\alpha]\varphi \qquad (Red_{+\alpha,\neg})$$

$$[-\alpha]\neg\varphi \leftrightarrow \begin{cases}\top & \text{if } \alpha \text{ is valid in } INTR\\ \neg[-\alpha]\varphi & \text{otherwise}\end{cases} \qquad (Red_{-\alpha,\neg})$$

$$[+\alpha](\varphi \wedge \varphi') \leftrightarrow [+\alpha]\varphi \wedge [+\alpha]\varphi' \qquad (Red_{+\alpha,\wedge})$$

$$[-\alpha](\varphi \wedge \varphi') \leftrightarrow [-\alpha]\varphi \wedge [-\alpha]\varphi' \qquad (Red_{-\alpha,\wedge})$$

$$[+\alpha]\beta \leftrightarrow \begin{cases}\top & \text{if } \alpha \rightsquigarrow \beta\\ \beta & \text{otherwise}\end{cases} \qquad (Red_{+\alpha})$$

$$[-\alpha]\beta \leftrightarrow \begin{cases}\top & \text{if } \alpha \text{ is valid in } INTR\\ \bot & \text{if } \alpha \text{ is not valid in } INTR \text{ and } \beta \rightsquigarrow \alpha\\ \beta & \text{otherwise}\end{cases} \qquad (Red_{-\alpha})$$

– the introspection axioms:

$$S_i\,S_i\,\alpha \qquad (Vis_{C1})$$

$$JS\;JS\,\alpha \qquad (Vis_{C2})$$

$$JS\;S_i\,S_i\,\alpha \qquad (Vis_{C3})$$

$$JS\,\alpha \to S_i\,\alpha \qquad (Vis_{C4})$$

$$JS\,\alpha \to JS\;S_i\,\alpha \qquad (Vis_{C5})$$

– the rule of Modus Ponens and the rules of inference for K_i, CK, and $[\pi]$:

$$\frac{\varphi \leftrightarrow \varphi'}{K_i\varphi \leftrightarrow K_i\varphi'} \qquad \frac{\varphi \leftrightarrow \varphi'}{CK\varphi \leftrightarrow CK\varphi'} \qquad \frac{\varphi \leftrightarrow \varphi'}{[\pi]\varphi \leftrightarrow [\pi]\varphi'}$$

Theorem 1. *The axiomatization of* DEL-PAO *is sound and complete.*

5 Complexity

Theorem 2. *The* DEL-PAO *satisfiability and* DEL-PAO *model checking problems are both* PSPACE*-complete.*

We devote the rest of the section to the proof of this result. We start by proving that all epistemic operators can be eliminated in polynomial time. We then show interreducibility of model and satisfiability checking. We finally establish lower and upper bounds by embedding QBF into DEL-PAO and DEL-PAO into DL-PA.

5.1 Elimination of epistemic operators

Let us define the following programs:

$$\pi_{i,\alpha} = S_i\,\alpha? \sqcup (\neg S_i\,\alpha?; (+\alpha \sqcup -\alpha))$$
$$\pi_{Agt,\alpha} = JS\,\alpha? \sqcup (\neg JS\,\alpha?; (+\alpha \sqcup -\alpha))$$

The first checks whether i sees α, and if not, varies the truth value of α; the second does the same but for joint visibility. Then for a set of atoms $A = \{\alpha_1, ..., \alpha_n\}$, we define:

$$\pi_{i,A} = \pi_{i,\alpha_1}; ...; \pi_{i,\alpha_n}$$
$$\pi_{Agt,A} = \pi_{Agt,\alpha_1}; ...; \pi_{Agt,\alpha_n}$$

We suppose that the program is $skip$ if the set A is empty.

We did not impose any ordering on atoms in A. One might wonder whether changing their truth values could affect the subsequent tests. However, this is not the case.

Proposition 5. *Let $\alpha, \beta \in ATM$ be two atoms. Then*

$$(S_i\,\alpha? \sqcup (\neg S_i\,\alpha?; (+\alpha \sqcup -\alpha))); (S_i\,\beta? \sqcup (\neg S_i\,\beta?; (+\beta \sqcup -\beta)))$$
$$\equiv (S_i\,\beta? \sqcup (\neg S_i\,\beta?; (+\beta \sqcup -\beta))); (S_i\,\alpha? \sqcup (\neg S_i\,\alpha?; (+\alpha \sqcup -\alpha))).$$

Proposition 6. *Let φ be a* DEL-PAO *formula without epistemic operators. Then*

$$K_i\varphi \leftrightarrow [\pi_{i,ATM(\varphi)}]\varphi$$
$$CK\varphi \leftrightarrow [\pi_{Agt,ATM(\varphi)}]\varphi$$

are valid in INTR.

Proposition 6 can be turned into a procedure eliminating epistemic operators: it suffices to iterate the application of the equivalences, starting with the innermost operators.

Procedure 1. *While there is an epistemic operator in φ:*

1. *if there exists a subformula $K_i\varphi'$ such that φ' does not contain epistemic operators, replace φ by $[\pi_{i,ATM(\varphi)}]\varphi'$;*
2. *if there exists a subformula $CK\varphi'$ such that φ' does not contain epistemic operators, replace φ by $[\pi_{Agt,ATM(\varphi)}]\varphi$.*

Proposition 7. *For every* DEL-PAO *formula φ, there exists a* DEL-PAO *formula φ' without epistemic operators such that $\varphi \leftrightarrow \varphi'$ is valid in INTR. The length of φ' is polynomial in $length(\varphi)$.*

5.2 Model checking and SAT interreducible

For formulas without epistemic operators, satisfiability and model checking have the same complexity.

Proposition 8. *Let φ be a* DEL-PAO *formula without epistemic operators such that $ATM(\varphi) = \{\alpha_1, \ldots, \alpha_n\}$. Let $\pi = (+\alpha_1 \sqcup -\alpha_1); \ldots; (+\alpha_n \sqcup -\alpha_n)$. Then:*

- *if φ is satisfiable in $INTR$, then for every $V \in INTR$, $V \models \langle\pi\rangle\varphi$;*
- *if φ is unsatisfiable in $INTR$, then for every $V \in INTR$, $V \not\models \langle\pi\rangle\varphi$.*

The length of the program $(+\alpha_1 \sqcup -\alpha_1); \ldots; (+\alpha_n \sqcup -\alpha_n)$ is linear in $length(\varphi)$. It follows from Proposition 8 that the satisfiability problem can be reduced in polynomial time to model checking in a randomly chosen valuation.

Proposition 9. *Let φ be a* DEL-PAO *formula without epistemic operators. For $V \in INTR$, $V \models \varphi$ if and only if the formula*

$$\langle +\alpha_1; \ldots; +\alpha_n; -\beta_1; \ldots; -\beta_m\rangle\varphi$$

is satisfiable in $INTR$, where $ATM(\varphi) \cap V = \{\alpha_1, \ldots, \alpha_n\}$ and $ATM(\varphi) \setminus V = \{\beta_1, \ldots, \beta_n\}$.

The length of $+\alpha_1; \ldots; +\alpha_n; -\beta_1; \ldots; -\beta_m$ is again linear in $length(\varphi)$. It follows from Proposition 9 that the model checking problem can be polynomially reduced to the satisfiability problem.

We observe that from a practical point of view, model checking requires a finite valuation. For formulas without epistemic operators such valuations can always be obtained: due to Proposition 4 we have $V \models \varphi$ iff $V \cap ATM(\varphi) \models \varphi$.

5.3 Lower bound

In DEL-PAO we can express Quantified Boolean Formulas (QBF), whose satisfiability problem is PSPACE-complete. Details can be found in the full version of the paper.

5.4 Dynamic Logic of Propositional Assignments

In order to establish the upper bound we will embed our logic into the star-free fragment of Dynamic Logic of Propositional Assignments DL-PA [12,4], whose satisfiability problem is PSPACE-complete. We briefly recall that logic.

Just as the language of DEL-PAO, the language of DL-PA has formulas and programs. They are defined by the following grammar:

$$\pi ::= +\alpha \mid -\alpha \mid \pi;\pi \mid \pi \sqcup \pi \mid \varphi?$$
$$\varphi ::= \alpha \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\pi]\varphi$$

where α ranges over ATM and i over Agt. So the language has the same atoms as DEL-PAO, but no epistemic operators.

The language is interpreted in valuations $V \in 2^{ATM}$ in exactly the same way as in DEL-PAO, except that the interpretation of atomic programs does not take introspective consequences into account. We have:

$$VR_{+\alpha}V' \text{ iff } V' = V \cup \{\alpha\}$$

$$VR_{-\alpha}V' \text{ iff } V' = V \setminus \{\alpha\}$$

A counterpart of Proposition 4 holds for DL-PA:

Proposition 10 ([4], Proposition 1). *Let $V, V' \in 2^{ATM}$ such that $V(\alpha) = V'(\alpha)$ for every atom $\alpha \in ATM(\varphi)$. Then $V \models_{\mathsf{DL\text{-}PA}} \varphi$ if and only if $V' \models_{\mathsf{DL\text{-}PA}} \varphi$.*

5.5 Upper bound

The final step is to polynomially translate non-epistemic DEL-PAO formulas and programs into DL-PA formulas and programs. The introspection constraints will be taken into account by translating DEL-PAO assignments into appropriate DL-PA programs.

Given an atom α and a set of relevant atoms $A \subseteq ATM$, let $Eff^+(\alpha) \cap A = \{\beta_1, ..., \beta_n\}$ and $Eff^-(\alpha) \cap A = \{\beta'_1, ..., \beta'_m\}$. Translate assignments of α as follows:

$$tr(+\alpha, A) = +\beta_1; ...; +\beta_n$$

$$tr(-\alpha, A) = \begin{cases} fail & \text{if } \alpha \text{ valid in } INTR \\ -\beta'_1; ...; -\beta'_m & \text{otherwise} \end{cases}$$

Again we suppose that the program is *skip* if the set $\{\beta_1, ..., \beta_n\}$ is empty.

We extend tr to complex programs and formulas by stipulating $tr(\alpha) = \alpha$ and $tr([\pi]\varphi) = [tr(\pi, ATM(\varphi))]tr(\varphi)$, and homomorphic otherwise.

Note that $ATM(tr(\pi, A)) \subseteq A$ and $ATM(tr(\varphi)) \subseteq ATM(\varphi)$.

Proposition 11. *Let φ be a DEL-PAO formula without epistemic operators. Then we have $V \models_{\mathsf{DEL\text{-}PAO}} \varphi$ if and only if $V \models_{\mathsf{DL\text{-}PA}} tr(\varphi)$.*

The grande finale follows from propositions 7, 8, 9 and 11 and because $tr(\varphi)$ can be computed in time polynomial in $length(\varphi)$.

Theorem 3. *In DEL-PAO, both satisfiability and model checking are PSPACE-complete.*

6 Private announcements and spreading gossip

Public Announcement Logic PAL [16] is a logic of the DEL family extending standard epistemic logic with an operator $[\psi!]$, such that $[\psi!]\varphi$ reads "after ψ is publicly and truthfully announced, φ is true". Its validities are axiomatized by means of the reduction axioms $[\psi!]p \leftrightarrow \psi{\rightarrow}p$, $[\psi!]\neg\varphi \leftrightarrow \psi{\rightarrow}\neg[\psi!]\varphi$, $[\psi!](\varphi \wedge \varphi') \leftrightarrow [\psi!]\varphi \wedge [\psi!]\varphi'$, and $[\psi!]K_i\varphi \leftrightarrow \psi{\rightarrow}K_i[\psi!]\varphi$.

We claim that we can express public announcements of literals as $p! = p?; +JS\,p$ and $\neg p! = \neg p?; +JS\,p$. We furthermore claim that we can express the public announcement of knowledge of atoms as $K_ip! = K_ip?; +JS\,p$. It can indeed be checked that with these definitions all the reduction axioms for PAL are valid in our logic (see the full version of the paper). Beyond that we can also easily model *private* announcements of the same kind of formulas. Read $j : \psi!$ as "ψ is privately announced to agent j". Then:

$$j : p! = p?; +S_j\,p$$
$$j : \neg p! = \neg p?; +S_j\,p$$

$$j : K_i p! = K_i p?; +S_j\, p; +S_j\, S_i\, p$$

Let us illustrate this by the Spreading Gossip problem, of which a detailed study can be found in [5]. The problem is also tackled in [17] with a logic specific to communication networks. Six friends each know a secret. When they call each other, they exchange every secret that they know. The problem is to find how many calls are necessary to spread all secrets among all friends. It was proven ([2], among others) that the minimal number of calls is 8; for example, if we write ij the fact that i calls j (or that j calls i), the following sequence spreads all secrets: 12, 34, 56, 13, 45, 16, 24, 35 [5].

Let us model this with private announcements. With $Agt = \{i : 1 \leq i \leq 6\}$ and s_i meaning that i has the secret s_i, we define the program $Call_{ij}$, for $i, j \in Agt$, as:

$$\begin{aligned} Call_{ij} = {} & ((S_i\, s_1?; j : s_1!) \sqcup \neg S_i\, s_1?); ...; ((S_i\, s_6?; j : s_6!) \sqcup \neg S_i\, s_6?); \\ & ((S_j\, s_1?; i : s_1!) \sqcup \neg S_j\, s_1?); ...; ((S_j\, s_6?; i : s_6!) \sqcup \neg S_j\, s_6?) \end{aligned}$$

Our program expresses that i tells all she knows to j, and conversely. So each call makes each atom known by one of the two agents known to both. Then the formula

$$\big[Call_{12}; Call_{34}; Call_{56}; Call_{13}; Call_{45}; Call_{16}; Call_{24}; Call_{35}\big] \bigwedge_{i \in Agt} K_i \left(\bigwedge_{j \in Agt} s_j \right)$$

is true at the initial state V_0 defined as:

$$V_0 = \{\alpha : \alpha \text{ is valid in } INTR\} \cup \{s_i : i \in Agt\} \cup \{S_i\, s_i : i \in Agt\}.$$

This establishes that the above sequence is correct. Furthermore, the formula

$$\langle (\bigsqcup_{i,j \in Agt, i \neq j} \neg S_i\, s_j?; Call_{ij})^8 \rangle \bigwedge_{i \in Agt} K_i \left(\bigwedge_{j \in Agt} s_j \right)$$

expresses that a more general protocol is correct. Finally, the formula

$$[(\bigsqcup_{i,j \in Agt, i \neq j} Call_{ij})^7] \neg \bigwedge_{i \in Agt} K_i \left(\bigwedge_{j \in Agt} s_j \right)$$

expresses that only 7 calls are not enough. Both are true at V_0.

Note that our modelling does not account for second-order knowledge. In order to do so we should modify the program $Call_{ij}$ in a way such that when $S_i\, s_1$ is true then not only $j : s_1!$ is performed, but also $i : S_j\, s_1!$. With that modelling we could check not only that everybody knows each secret, but also that everybody knows that everybody knows each secret. In the same vein, third-order knowledge can be attained by adding $j : S_i\, S_j\, s_1!$, and so on.

Beyond that, we may also want to model that $Call_{ij}$ leads to common knowledge of i and j. This requires the extension of DEL-PAO by visibility atoms with non-empty sets of agents as arguments. However, secrets can never become common knowledge of *all* agents. This can also be highlighted by the Two Generals' problem [1,9] where common knowledge cannot be reached. Details are in the full version of the paper.

7 Related work

As said in the introduction, our logic is in the tradition of several other logics developed in the past few years. In the logic ECL-PC(PO) [14], visibility is represented by a set of atoms for each agent, containing the variables the agent observes. This does not allow for higher-order observations such as "i observes whether j observes p". Instead and as already mentioned, the observational capabilities of each agent become common knowledge among all agents.

The logic LRC [13] allows to express, as programs, that a variable is revealed to an agent or concealed from her. Semantically, formulas are interpreted over pointed models with a visibility set for each agent; revealing a variable p to an agent i will add p to i's visibility set, while concealing p will remove p from i's set. Just as in ECL-PC(PO), who sees what is common knowledge among all agents.

The logic of knowing whether [7] adds an operator standing for "i knows whether φ" to the language of standard epistemic logic, interpreted as "φ has the same value in all indistinguishable worlds for i". This can be compared to our visibility atoms S_i which express the same notion on atoms.

In Flatland Logic [3], visibility is further grounded on geometry in order to give semantics to epistemic operators: an agent can (or cannot) observe the positions of other agents and can reason about what they observe. Visibility can be higher-order and is also fully determined by geometric constraints. The main difference with our logic is that in Flatland Logic, agents see other agents instead of propositional variables.

8 Conclusion

We have introduced a dynamic epistemic logic of propositional assignment and observation DEL-PAO which accounts for higher-order and joint observation as well as updates thereof. It avoids the strong hypothesis of common knowledge of visibility that other observation-based epistemic logics make. It is remarkable that the addition of higher-order observability and in particular of joint observability comes without supplementary cost: both satisfiability and model checking remain PSPACE-complete. This contrasts with standard logics of common knowledge: there, satisfiability checking is EXPTIME-hard [10].

A simple extension of our logic is to generalize the operator of common knowledge of all agents CK to operators taking any subset of Agt as arguments. It suffices to introduce visibility atoms $JS_J\alpha$, one per group of agents J. Another interesting generalization is to consider belief instead of knowledge. A way to achieve this is to replace S_i by two operators O_i and C_i, respectively meaning that i has an opinion on something and that i is correct on something. This requires other constraints on valuations that should match the properties of belief. Further possible extensions concern the dynamic part: following [12], one may add atoms representing that i controls some propositional variable p, in the sense that i can change the truth value of p at will. One may then associate to each assignment an author, which is the agent performing the assignment. As shown in [12], this allows to embed Coalition Logic of Propositional Control [14]. It remains to be worked out how this combines with higher-order observations.

Acknowledgments

We would like to thank François Schwarzentruber for useful comments and the anonymous reviewers for their thoughtful reading and comments.

References

1. Akkoyunlu, E.A., Ekanadham, K., Hubert, R.V.: Some constraints and tradeoffs in the design of network communications. In: Proceedings of the 5th ACM Symposium on Operating Systems Principles. pp. 67–74. ACM Press (1975)
2. Baker, B., Shostak, R.: Gossips and telephones. Discrete Mathematics 2(3), 191–193 (1972)
3. Balbiani, P., Gasquet, O., Schwarzentruber, F.: Agents that look at one another. Logic Journal of the IGPL 21(3), 438–467 (2013)
4. Balbiani, P., Herzig, A., Troquard, N.: Dynamic logic of propositional assignments: a well-behaved variant of PDL. In: Kupferman, O. (ed.) Proceedings of the 28th Annual IEEE/ACM Symposium on Logic in Computer Science. pp. 143–152 (2013)
5. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Springer Publishing Company, Incorporated, 1st edn. (2007)
6. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning about Knowledge. MIT Press (1995)
7. Fan, J., Wang, Y., van Ditmarsch, H.: Knowing whether. CoRR abs/1312.0 (2013)
8. Fischer, M.J., Ladner, R.E.: Propositional dynamic logic of regular programs. Journal of Computer and System Sciences 18(2), 194–211 (1979)
9. Gray, J.: Notes on data base operating systems. In: Operating Systems, An Advanced Course. pp. 393–481. Springer-Verlag (1978)
10. Halpern, J.Y., Moses, Y.: A guide to completeness and complexity for modal logics of knowledge and belief. Artificial Intelligence 54(3), 319–379 (1992)
11. Herzig, A.: Logics of knowledge and action: critical analysis and challenges. Journal of Autonomous Agents and Multi-Agent Systems pp. 1–35 (2014), online July 2, 2014, doi : 10.1007/s10458-014-9267-z.
12. Herzig, A., Lorini, E., Troquard, N., Moisan, F.: A dynamic logic of normative systems. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence. pp. 228–233 (2011)
13. van der Hoek, W., Iliev, P., Wooldridge, M.: A logic of revelation and concealment. In: van der Hoek, W., Padgham, L., Conitzer, V., Winikoff, M. (eds.) Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems. pp. 1115–1122. IFAAMAS (2012)
14. van der Hoek, W., Troquard, N., Wooldridge, M.: Knowledge and control. In: Sonenberg, L., Stone, P., Tumer, K., Yolum, P. (eds.) Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems. pp. 719–726. IFAAMAS (2011)
15. Lorini, E., Herzig, A.: Direct and indirect common belief. In: Institutions, Emotions, and Group Agents, pp. 355–372. Springer Netherlands (2014)
16. Plaza, J.: Logics of public communications. In: Emrich, M.L., Pfeifer, M.S., Hadzikadic, M., Ras, Z. (eds.) Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems. pp. 201–216. Oak Ridge National Laboratory, ORNL/DSRD- 24 (1989)
17. Wang, Y., Sietsma, F., van Eijck, J.: Logic of information flow on communication channels. In: Omicini, A., Sardiña, S., Vasconcelos, W.W. (eds.) DALT. pp. 130–147. Springer (2010)