Building Epistemic Logic from Observations and Public Announcements

T. Charrier[1] and **A. Herzig**[2] and **E. Lorini**[2] and **F. Maffre**[2] and **F. Schwarzentruber**[1]
[1]IRISA, ENS Rennes, France
[2]IRIT, University of Toulouse, France

Abstract

We study an epistemic logic where knowledge is built from what the agents observe (including higher-order visibility) and what the agents learn from public announcements. This fixes two main drawbacks of previous observability-based approaches where who sees what is common knowledge and where the epistemic operators distribute over disjunction. The latter forbids the modeling of most of the classical epistemic problems, starting with the muddy children puzzle. We integrate a dynamic dimension where both facts of the world and the agents' observability can be modified by assignment programs. We establish that the model checking problem is PSPACE-complete.

Keywords: epistemic logic, dynamic epistemic logic, public announcement, propositional assignment, propositional observability

Introduction

Recently several logics based on the observability of propositional variables by agents were proposed (van der Hoek, Troquard, and Wooldridge 2011; Balbiani, Gasquet, and Schwarzentruber 2013; Gasquet, Goranko, and Schwarzentruber 2014). The models of these logics are based on visibility atoms from which Kripkean accessibility relations can be defined in a natural way: agent i cannot distinguish w from w' when all the variables agent i sees have the same truth value at w and w'. Such logics are attractive because their models are much more compact than the standard Kripke models of epistemic logics. The existing approaches however have two major drawbacks. First, who sees what is common knowledge: the formula $(\mathtt{K}_i p \vee \mathtt{K}_i \neg p) \rightarrow \mathtt{CK}(\mathtt{K}_i p \vee \mathtt{K}_i \neg p)$ is valid. Second, the epistemic operator may distribute over disjunctions: when p and q are different propositional variables then $\mathtt{K}_i(p \vee q) \rightarrow (\mathtt{K}_i p \vee \mathtt{K}_i q)$ is valid. The former is annoying because theory of mind is 'flattened'. The latter is annoying because it does not allow to model things such as the muddy children puzzle (where each child knows that one of the children is muddy without knowing which) (Lehmann 1984; Fagin et al. 1995). In the present paper we propose a solution to these two problems.

As to the first problem, we adopt the solution of (Herzig, Lorini, and Maffre 2015) and suppose that the valuations contain *higher-order visibility atoms*. Such atoms may take for example the form $\mathtt{JS}\,\mathtt{S}_j\, p$, expressing that all agents jointly see whether agent j sees the value of the propositional variable p. From this we obtain accessibility relations that are not common knowledge.

As to the second problem, following (Castelfranchi 1994; van Linder, Hoek, and Meyer 1997), we start by observing that an agent's knowledge may originate from three processes: observation, communication, and inference. We do not consider knowledge obtained via inference and assume that agents are omniscient. We therefore do not model formation of knowledge via (time-consuming) application of inference rules and leave it to future work to integrate existing logics of time-bounded reasoning (Alechina, Logan, and Whitsey 2004; Grant, Kraus, and Perlis 2000; Balbiani, Fernandez-Duque, and Lorini 2016). While the above observability-based approaches only account for the former, we here take into account the most basic form of communication modifying the agents' knowledge: public announcements. We do so by adding a *public information state* to the model, which is a set of valuations, as proposed in (Lomuscio, van der Meyden, and Ryan 2000; Su, Sattar, and Luo 2007) and recently used in (Charrier and Schwarzentruber 2015; van Benthem et al. 2015). Public announcements make the public information state shrink just as in public announcement logic. We can then model that although both a p-world and a q-world are accessible for i, there is no accessible $p \wedge q$-world because the information state contains no such world. This may be due to the announcement of $p \wedge q$, or to the announcement (in some order) of p and of q, or to the announcement (in some order) of p and of $p \rightarrow q$, etc.

Our approach is based on programs of Dynamic Logic of Propositional Assignments DL-PA (Herzig et al. 2011; Balbiani, Herzig, and Troquard 2013), which is a dialect of propositional dynamic logic whose atomic programs are $+p$ and $-p$, for p being a propositional variable: the program $+p$ makes p true and $-p$ makes p false. Complex programs are built from these atomic programs by the PDL program operators. We view such programs as being executed publicly, thereby updating the public information state. Public announcements will be a particular case of such publicly executed programs: they are publicly executed tests. Our program-based approach to knowledge and its dynamics also allows to modify visibility information, viz. by means of the assignment of visibility atoms to true or false. For example, the program $+\mathtt{S}_i\,\mathtt{S}_j\,p$ has the effect that i sees whether j sees p.

The paper is organized as follows: we first present the language then the semantics. Then we establish the complexity of the model checking problem and study the properties of our logic. Finally, we illustrate our logic by a full analysis of the muddy children puzzle and conclude.

Language

Let $Prop$ be a countable non-empty set of propositional variables and let Agt be a finite non-empty set of agents.

Observability atoms

Atomic formulas of our language are sequences of visibility operators followed by propositional variables. For instance, $\mathtt{JS}\,\mathtt{S}_1\,p$ is an atomic formula. It is read "all agents jointly see whether agent 1 sees the truth value of p".

Formally, the set of *observability operators* is

$$OBS = \{\mathtt{S}_i : i \in Agt\} \cup \{\mathtt{JS}\},$$

where $\mathtt{S}_i$ stands for individual visibility of agent i and $\mathtt{JS}$ 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, observability atoms are propositional variables prefixed by a possibly empty sequence of observability operators. So an observability atom is of the form $\sigma\, p$ where $p \in Prop$ and $\sigma \in OBS^*$. We use $\alpha, \alpha', \ldots$ for observability atoms. So propositional variables are particular observability atoms.

Programs and formulas

The language of programs and formulas is defined by the following grammar:

$$\pi ::= +\alpha \mid -\alpha \mid (\pi;\pi) \mid (\pi \sqcup \pi) \mid \varphi?$$
$$\varphi ::= \alpha \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \mathtt{K}_i\varphi \mid \mathtt{CK}\varphi \mid [\pi\,!]\varphi$$

where α ranges over ATM and i over Agt.

As to the formulas: $\mathtt{K}_i\varphi$ reads "φ is known by i" and $\mathtt{CK}\varphi$ reads "φ is common knowledge among all agents". While $\mathtt{S}_i$ expresses sensor information, $\mathtt{K}_i$ expresses information coming from both sensors and communication. We read $[\pi\,!]\varphi$ as "φ will be true after the update of the current local and information states by π".

As to the programs: $+\alpha$ makes the atomic formula α publicly true and $-\alpha$ makes it publicly false. We will see in the semantics that $-\alpha$ may fail. For instance, $\mathtt{S}_i\,\mathtt{S}_i\,\alpha$ is always true (agents are introspective) and the execution of $-\mathtt{S}_i\,\mathtt{S}_i\,\alpha$ fails. The other program operators are sequential and nondeterministic composition and test and are read as in dynamic logic (Harel, Kozen, and Tiuryn 2000). As we shall see, the publicly performed test $\varphi?$ behaves exactly as the public announcement of φ.

As usual in modal logic, $\langle\pi!\rangle\varphi$ abbreviates $\neg[\pi\,!]\neg\varphi$.

Example 1 (representing the muddy children puzzle)**.** Let us illustrate by means of the muddy children puzzle how knowledge can be represented in our language. Let $Agt = \{1, \ldots, n\}$ be the set of children. Let $Prop = \{m_1, \ldots, m_n\}$ be the propositional variables, where m_i expresses that child i is muddy. First, the fact that all children are muddy is described by the conjunction

$$Muddy = \bigwedge_{i \in Agt} m_i.$$

Second, agents' observational capabilities are expressed by

$$Obs = \Big(\bigwedge_{i\in Agt}\neg\mathtt{S}_i\,m_i\Big) \wedge \Big(\bigwedge_{\substack{i,j\in Agt,\\ i\neq j}}\mathtt{S}_i\,m_j\Big) \wedge \Big(\bigwedge_{i,j\in Agt}\mathtt{JS}\,\mathtt{S}_i\,m_j\Big).$$

The first two conjuncts express that the agents see the states of other agents but not their own states. The last conjunct expresses that the agents jointly observe their observational capabilities. For instance, $\mathtt{JS}\,\mathtt{S}_i\,m_i$ reads "all agents jointly see whether i sees m_i". Third, the agents' (sensor and communicational) information in the initial situation is described by the formula

$$Ign = \bigwedge_{i\in Agt}(\neg\mathtt{K}_i m_i \wedge \neg\mathtt{K}_i\neg m_i).$$

So the muddy children puzzle is fully described by the conjunction $Muddy \wedge Obs \wedge Ign$.

Let us now look at the consequences of this description in the semantics to be defined. First, the implication

$$(Muddy \wedge Obs \wedge Ign) \to \langle(\bigvee_{i\in Agt} m_i)?!\rangle Ign$$

will be valid for $n \geq 2$: each child is still ignorant about her muddiness after the announcement that one of them is muddy. Furthermore, the implications

$$(Muddy \wedge Obs \wedge Ign) \to \langle(\bigvee_{i\in Agt} m_i)?!\rangle\langle Ign?!\rangle^k Ign$$

will be valid for $0 \leq k \leq n{-}2$ and $n \geq 2$, where $\langle\pi!\rangle^k$ is the iteration of $\langle\pi!\rangle$, k times: the children keep on being ignorant about their state after $n{-}2$ rounds of the announcement of that ignorance. It is important to note that the intended meaning of the formulas $\neg\mathtt{K}_i m_i \wedge \neg\mathtt{K}_i\neg m_i$ and $\neg\mathtt{S}_i\,m_i$ is different: the former says that according to her information state, i is ignorant about m_i, while the latter says that according to her observational information, i does not observe m_i. The status of the latter remains unchanged when the children gain new information via the public announcement of Ign. In contrast, the status of the former changes after $n{-}1$ announcements:

$$(Muddy \wedge Obs \wedge Ign) \to \langle(\bigvee_{i\in Agt} m_i)?!\rangle\langle Ign?!\rangle^{n-1}(\bigvee_{i\in Agt}\mathtt{K}_i m_i)$$

and even

$$(Muddy \wedge Obs \wedge Ign) \to \langle(\bigvee_{i\in Agt} m_i)?!\rangle\langle Ign?!\rangle^{n-1}\mathtt{CK}(\bigvee_{i\in Agt} m_i)$$

will be valid: knowledge and even common knowledge of muddiness is achieved after $n-1$ rounds.

Semantics

Our semantics is based on valuations, that we will also call worlds. These are nothing but subsets on the set of atomic formulas ATM, noted $w, w', u, v, \ldots$. We write $w(\alpha) = w'(\alpha)$ when atomic formula α has the same truth value in w and w', i.e., when either both $\alpha \in w$ and $\alpha \in w'$, or both $\alpha \notin w$ and $\alpha \notin w'$.

Accessibility relations for $\mathtt{K}_i$ and $\mathtt{CK}$

From the visibility information that is contained in valuations we are now going to define accessibility relations between valuations. Two valuations w and w' are related by $\rightsquigarrow_i$ if every α that i sees at w has the same value, and similarly for the common knowledge accessibility relation $\rightsquigarrow_{Agt}$:

$$w \rightsquigarrow_i w' \quad \text{iff} \quad \mathtt{S}_i\,\alpha \in w \text{ implies } w(\alpha) = w'(\alpha)$$
$$w \rightsquigarrow_{Agt} w' \quad \text{iff} \quad \mathtt{JS}\,\alpha \in w \text{ implies } w(\alpha) = w'(\alpha)$$

In logics of knowledge these relations are standardly supposed to be equivalence relations. This fails to hold here: while $\rightsquigarrow_i$ and $\rightsquigarrow_{Agt}$ are clearly reflexive, they are neither transitive nor symmetric.[1] We will see in the sequel how transitivity and symmetry can be guaranteed by means of appropriate introspection constraints.

Introspective valuations

A valuation $w \in 2^{ATM}$ is *introspective* if and only if the following hold, for every $\alpha \in ATM$ and $i \in Agt$:

$$\mathtt{S}_i\,\mathtt{S}_i\,\alpha \in w \qquad \text{(C1)}$$
$$\mathtt{JS}\,\mathtt{JS}\,\alpha \in w \qquad \text{(C2)}$$
$$\mathtt{JS}\,\mathtt{S}_i\,\mathtt{S}_i\,\alpha \in w \qquad \text{(C3)}$$
$$\text{if } \mathtt{JS}\,\alpha \in w, \text{ then } \mathtt{S}_i\,\alpha \in w \qquad \text{(C4)}$$
$$\text{if } \mathtt{JS}\,\alpha \in w, \text{ then } \mathtt{JS}\,\mathtt{S}_i\,\alpha \in w \qquad \text{(C5)}$$

For example, 2^{ATM} is introspective, while $\emptyset$ and $\{p, q\}$ are not. The set of all introspective valuations is noted $INTR$. Let us stress that neither $\mathtt{S}_i\,\alpha \in w$ nor $\mathtt{JS}\,\alpha \in w$ imply $\alpha \in w$: seeing *whether* α is true is weaker than seeing *that* α is true.

Together, the last two constraints ensure that when $\mathtt{JS}\,\alpha \in w$ then $\sigma\,\alpha \in w$ for every $\sigma \in OBS^+$. This motivates the following relation of *introspective consequence* between atoms:

$$\alpha \Rightarrow \beta \quad \text{iff} \quad \alpha = \beta, \text{ or } (\alpha = \mathtt{JS}\,\alpha' \text{ and } \beta = \sigma\,\alpha' \text{ for some } \sigma \in OBS^+).$$

For example, $\mathtt{JS}\,\mathtt{S}_i\,p \Rightarrow \mathtt{S}_{j_1}\,\mathtt{S}_{j_2}\,\mathtt{S}_i\,p$: when all agents jointly see whether i observes the value of p then j_1 sees whether j_2 sees whether i sees p.

Introspective valuations can be characterized as valuations that satisfy (C1), (C2) and (C3) and are closed under introspective consequence.

[1]For example, $\emptyset \rightsquigarrow_i w$ for every $w \subseteq ATM$, while $w \rightsquigarrow_i \emptyset$ fails to hold as soon as there is a p such that p and $\mathtt{S}_i\,p$ are in w.

Proposition 1 ((Herzig, Lorini, and Maffre 2015))**.** *A valuation $w \subseteq ATM$ is introspective if and only if, for every $\alpha \in ATM$ and $i \in Agt$:*

$$\sigma\,\mathtt{S}_i\,\mathtt{S}_i\,\alpha \in w \text{ for every } \sigma \in OBS^* \qquad (1)$$
$$\sigma\,\mathtt{JS}\,\alpha \in w \text{ for every } \sigma \in OBS^+ \qquad (2)$$
$$\text{if } \alpha \in w \text{ and } \alpha \Rightarrow \beta \text{ then } \beta \in w \qquad (3)$$

We say that an atom α 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 $\sigma\,\mathtt{S}_i\,\mathtt{S}_i\,\alpha$ with $\sigma \in OBS^*$, or $\sigma\,\mathtt{JS}\,\alpha$ with $\sigma \in OBS^+$.

Proposition 2 ((Herzig, Lorini, and Maffre 2015))**.** *The relation $\rightsquigarrow_{Agt}$ and every relation $\rightsquigarrow_i$ are equivalence relations on $INTR$.*

Let us illustrate this by means of two muddy children.

Example 2 (two muddy children)**.** Consider the valuation

$$w = \{\mathtt{S}_1\,m_2, \mathtt{S}_2\,m_1\} \cup \{\alpha : \alpha \text{ is valid in } INTR\} \cup \{\sigma \mathtt{S}_i\,m_j : \sigma \in OBS^+, i, j \in \{1,2\}\}.$$

The last two lines ensure that w is introspective: the second adds all introspective validities and the third adds $\mathtt{JS}\,\mathtt{S}_i\,m_j$ and all its introspective consequences. Then the four relevant introspective valuations are depicted in Figure 1, together with relations $\rightsquigarrow_1$ and $\rightsquigarrow_2$.

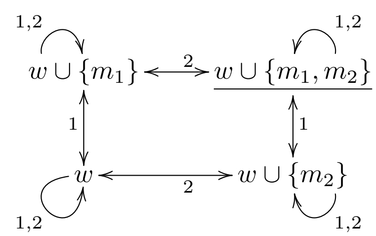

Figure 1: Two muddy children.

Truth conditions and validity

Given an introspective valuation, our update operations add or remove atoms from it. This requires some care: we want the resulting valuation to be introspective, too. For example, removing $\mathtt{S}_i\,\mathtt{S}_i\,p$ should be impossible. Another example is when the introspective valuation w does not contain $\mathtt{S}_i\,p$: then $w \cup \{\mathtt{JS}\,p\}$ would violate (C4). In order to avoid this, when adding an atom to w we also add all its *positive consequences*. Symmetrically, when removing an atom we also have to remove its *negative consequences*. To that end, let us define the following update operations on valuations:

$$w{+}\alpha = w \cup \{\beta \in ATM : \alpha \Rightarrow \beta\}$$
$$w{-}\alpha = w \setminus \{\beta \in ATM : \beta \Rightarrow \alpha\}$$

When w is introspective then both $w{+}\alpha$ and $w{-}\alpha$ are so, too (the latter with the proviso that α is not valid in $INTR$). We extend the definition to updates of sets of valuations $U \subseteq 2^{ATM}$ in the obvious way: $U{+}\alpha = \{w{+}\alpha : w \in U\}$ and $U{-}\alpha = \{w{-}\alpha : w \in U\}$.

A *pointed model* is a couple $\langle U, w\rangle$ where $U \subseteq 2^{ATM}$ and $w \subseteq ATM$. We call U the information state and w the local state. Note that we do not require $w \in U$. In these models formulas are interpreted as follows:

$$
\begin{array}{lll}
U, w \models \alpha & \text{iff} & \alpha \in w \\
U, w \models \neg\varphi & \text{iff} & \text{not } (U, w \models \varphi) \\
U, w \models \varphi \wedge \varphi' & \text{iff} & U, w \models \varphi \text{ and } U, w \models \varphi' \\
U, w \models [\pi\,!]\varphi & \text{iff} & U', w' \models \varphi \text{ for every } \langle U', w'\rangle \\
 & & \quad \text{such that } \langle U, w\rangle \mathcal{P}_\pi \langle U', w'\rangle \\
U, w \models \mathtt{K}_i\varphi & \text{iff} & U, w' \models \varphi \text{ for every } w' \in U \\
 & & \quad \text{such that } w \rightsquigarrow_i w' \\
U, w \models \mathtt{CK}\varphi & \text{iff} & U, w' \models \varphi \text{ for every } w' \in U \\
 & & \quad \text{such that } w \rightsquigarrow_{Agt} w'
\end{array}
$$

where $\mathcal{P}_\pi$ is the relation of public update on pointed models defined as follows:

$$
\begin{array}{lll}
\langle U, w\rangle \mathcal{P}_{+\alpha} \langle U', w'\rangle & \text{iff} & U' = U{+}\alpha \text{ and } w' = w{+}\alpha \\
\langle U, w\rangle \mathcal{P}_{-\alpha} \langle U', w'\rangle & \text{iff} & U' = U{-}\alpha \text{ and } w' = w{-}\alpha \\
 & & \quad \text{and } \alpha \text{ is not valid in } INTR \\
\langle U, w\rangle \mathcal{P}_{\pi;\pi'} \langle U', w'\rangle & \text{iff} & \langle U, w\rangle (\mathcal{P}_\pi \circ \mathcal{P}_{\pi'}) \langle U', w'\rangle \\
\langle U, w\rangle \mathcal{P}_{\pi\sqcup\pi'} \langle U', w'\rangle & \text{iff} & \langle U, w\rangle (\mathcal{P}_\pi \cup \mathcal{P}_{\pi'}) \langle U', w'\rangle \\
\langle U, w\rangle \mathcal{P}_{\chi?} \langle U', w'\rangle & \text{iff} & U, w \models \chi,\ w'{=}w, \text{ and} \\
 & & \quad U' = \{u \in U : U, u \models \chi\}
\end{array}
$$

For example, both $\neg\mathtt{S}_i\, p \wedge \neg\mathtt{S}_i\, q$ and $\mathtt{K}_i(p \leftrightarrow q)$ are true in the pointed model $\langle\{\emptyset, \{p, q\}\}, \emptyset\rangle$.

Example 3 (two muddy children, continued)**.** Let U be the set of all valuations in Figure 1. We have:

$$
\begin{array}{r}
U, u \models Obs \wedge Ign, \text{ for every } u \in U \\
U, w{\cup}\{m_1, m_2\} \models \langle m_1{\vee}m_2?!\rangle Ign \\
U, w{\cup}\{m_1, m_2\} \models \langle m_1{\vee}m_2?!\rangle\langle Ign?!\rangle(\mathtt{K}_1 m_1 \wedge \mathtt{K}_2 m_2)
\end{array}
$$

Let $\mathcal{C}$ be a class of pointed models. A formula φ is *satisfiable in* $\mathcal{C}$ if and only if there is a $\langle U, w\rangle \in \mathcal{C}$ such that $U, w \models \varphi$; it is *valid in* $\mathcal{C}$ if and only if $\neg\varphi$ is unsatisfiable. For example, the equivalence $[\chi?\,!]\bot \leftrightarrow \neg\chi$ is valid in the class of all pointed models.

A *pointed epistemic model* is a pointed model $\langle U, w\rangle$ such that $U \subseteq INTR$ and $w \in U$. For example, $\mathtt{S}_i\ \mathtt{JS}\, p$ is valid in epistemic models. The following conditions guarantee that when we interpret a formula in a pointed epistemic model we stay within the class of pointed epistemic models.

Proposition 3. *Let $\langle U, w\rangle$ be a pointed epistemic model. Then the following hold:*

1. *If $w \rightsquigarrow_i w'$ and $w' \in U$ then $\langle U, w'\rangle$ is a pointed epistemic model.*
2. *If $w \rightsquigarrow_{Agt} w'$ and $w' \in U$ then $\langle U, w'\rangle$ is a pointed epistemic model.*
3. *If $\langle U, w\rangle \mathcal{P}_\pi \langle U', w'\rangle$ then $\langle U', w'\rangle$ is a pointed epistemic model.*

Observe that the last item does not require that $w' \in U'$: this is guaranteed by the fact that $\langle U, w\rangle$ is a pointed epistemic model and the definition of $\mathcal{P}_\pi$.

It is important to observe that the schemas

$$
\mathtt{S}_i\, \alpha \wedge \alpha \rightarrow \mathtt{K}_i\alpha
$$
$$
\mathtt{S}_i\, \alpha \wedge \neg\alpha \rightarrow \mathtt{K}_i\neg\alpha
$$

are valid in the class of all pointed models (even non-epistemic ones). In contrast, the converse of the implication is invalid. For example, for $U = \{w \in 2^{ATM} : p \in w\}$ we have $\langle U, \{p\}\rangle \models \mathtt{K}_i p$ while $\langle U, \{p\}\rangle \not\models \mathtt{S}_i\, p$.

Expressing public announcement of formulas

Consider the operator $[\chi\,!]$ of public announcement of a formula χ as studied in dynamic epistemic logics (van Ditmarsch, van der Hoek, and Kooi 2007). In the present setting, its truth condition has to be formulated as follows:

$$
U, w \models [\chi\,!]\varphi \text{ iff } U, w \models \chi \text{ implies } \{u \in U : U, u \models \chi\}, w \models \varphi.
$$

The set $\{u \in U : U, u \models \chi\}$ is called the relativization of U to the extension of χ in U.

Let us compare this to the public performance of tests: the relativization of U to the extension of χ in U is nothing but the result of the public update of U by $\chi?$. Indeed, $\langle U, w\rangle \mathcal{P}_{\chi?} \langle U', w'\rangle$ is the case if and only if $U, w \models \chi$ and U' is the restriction of U to the extension of χ in U. So $[\chi\,!]\varphi$ and $[\chi?\,!]\varphi$ have identical truth conditions.

Adding mental programs

It will be useful for the sequel to extend our language of formulas by adding a further modal operator $\mathtt{K}_\pi$, where π is viewed as a *mental program*. Such operators were introduced e.g. in (van Benthem, van Eijck, and Kooi 2006; Charrier and Schwarzentruber 2015; Herzig, Lorini, and Maffre 2015). While $[\pi\,!]\varphi$ is read "φ will be true after the public update of the current local and information states by π", we read $\mathtt{K}_\pi\varphi$ as "φ will be true after the update of the current local state by π (keeping the current information state constant)". So at a given local state w and information state U, the public program operator $[\pi\,!]$ updates both U and w; in contrast, the mental program operator $\mathtt{K}_\pi$ keeps U constant and only updates w. So the latter can be viewed as traversing the space of current epistemic possibilities.

Let us give an example: suppose there are only three propositional variables p, q, r and suppose agent i sees p but neither q nor r. This can then be modeled by the program $(+q \sqcup -q); (+r \sqcup -r)$ that nondeterministically changes q and r while keeping p unchanged. Indeed, the accessibility relation that we are going to define for $(+q{\sqcup}{-q}); (+r{\sqcup}{-r})$ coincides with $\rightsquigarrow_i$ in the language whose only variables are p, q, r. Visibility information therefore allows to reduce epistemic operators to mentally executed programs.

Interpreting mental programs

In the formula $\mathtt{K}_\pi\varphi$, both the program π and the formula φ may contain some $\mathtt{K}_{\pi'}$. So the precise definition of the language is recursive.

The semantics of the additional operator $\mathtt{K}_\pi$ is as follows:

$$
\begin{array}{lll}
U, w \models \mathtt{K}_\pi\varphi & \text{iff} & U, w' \models \varphi \text{ for every } w' \in U \\
 & & \quad \text{such that } \langle U, w\rangle \mathcal{M}_\pi \langle U, w'\rangle
\end{array}
$$

where $\mathcal{M}_\pi$ is the mental relation on pointed models that is defined as follows:

$$\begin{array}{ll} \langle U, w\rangle \mathcal{M}_{+\alpha} \langle U', w'\rangle & \text{iff } U'{=}U \text{ and } w' = w{+}\alpha \\ \langle U, w\rangle \mathcal{M}_{-\alpha} \langle U', w'\rangle & \text{iff } U'{=}U \text{ and } w' = w{-}\alpha \\ & \quad \text{and } \alpha \text{ is not valid in } \mathit{INTR} \\ \langle U, w\rangle \mathcal{M}_{\pi;\pi'} \langle U', w'\rangle & \text{iff } \langle U, w\rangle (\mathcal{M}_\pi \circ \mathcal{M}_{\pi'}) \langle U', w'\rangle \\ \langle U, w\rangle \mathcal{M}_{\pi \sqcup \pi'} \langle U', w'\rangle & \text{iff } \langle U, w\rangle (\mathcal{M}_\pi \cup \mathcal{M}_{\pi'}) \langle U', w'\rangle \\ \langle U, w\rangle \mathcal{M}_{\chi ?} \langle U', w'\rangle & \text{iff } U'{=}U, w'{=}w \text{ and } U, w \models \chi \end{array}$$

Observe that mental programs do not change the information state: when $\langle U, w\rangle \mathcal{M}_\pi \langle U', w'\rangle$ then $U' = U$. Observe also that mental programs can exit the public information state U during their execution. However, our truth condition for $\mathtt{K}_\pi$ requires $w' \in U$: we only take into account executions of π terminating inside the public information state U.

Let us illustrate the difference between the relations of public and mental program execution by an example.

Example 4. Let $U = \{w \in 2^{ATM} : p \notin w\}$. Then $\mathcal{M}_{+p}$ relates the pointed epistemic model $\langle U, \emptyset\rangle$ to the model $\langle U, \{p\}\rangle$, but $\{p\} \notin U$. Therefore $\langle U, \emptyset\rangle \models \mathtt{K}_{+p}\bot$.

In contrast, $\mathcal{P}_{+p}$ relates $\langle U, \emptyset\rangle$ to $\langle U{+}p, \{p\}\rangle$, where $U{+}p = \{w \in 2^{ATM} : p \in w\}$. Therefore $\langle U, \emptyset\rangle \not\models [+p\,!]\bot$.

From epistemic operators to mental programs

In this section we show how to encode epistemic operators by means of mental programs. Consider the following example:

$$\{\emptyset, \{p, q\}\}, \emptyset \models \mathtt{K}_i(p \leftrightarrow q)$$
$$\{\emptyset, \{p, q\}\}, \emptyset \models \mathtt{K}_{(+p \sqcup -p);(+q \sqcup -q)}(p \leftrightarrow q)$$

This illustrates that $\mathtt{K}_i$ and $\mathtt{K}_{(+p \sqcup -p);(+q \sqcup -q)}$ are evaluated in the same way as long as we consider formulas built with p and q. Let us give a more general example.

Example 5. Suppose agent i's mental program is

$$\pi = (\mathtt{S}_i\, p? \sqcup (\neg \mathtt{S}_i\, p?; (+p \sqcup -p))); (\mathtt{S}_i\, q? \sqcup (\neg \mathtt{S}_i\, q?; (+q \sqcup -q)))$$

The program π computes the worlds that are possible for agent i by varying the value of p if i does not see p, and similarly for q. When U contains the valuation $\{p, q\}$ then the latter is a possible valuation for agent i at the current valuation $\emptyset$. Indeed, $(\emptyset, \{p\}, \{p, q\})$ is a possible trace of π.

Now suppose that the formula $p \leftrightarrow q$ is announced. Then $\{p, q\}$ should still be a possible valuation for agent i at $\emptyset$ because $U, \{p, q\} \models p \leftrightarrow q$ for every U. That valuation is indeed still accessible via the above trace because we only make sure that we are in the public information state at the endpoint of the trace.

Let us generalize the above examples. The following programs will be helpful:

$$\begin{aligned} \mathsf{varyIfNotSeen}(i, \alpha) &= \mathtt{S}_i\, \alpha? \sqcup (\neg \mathtt{S}_i\, \alpha?; (+\alpha \sqcup -\alpha)) \\ \mathsf{varyIfNotSeen}(i, A) &= \mathsf{varyIfNotSeen}(i, \alpha_1); \cdots ; \\ &\quad\ \mathsf{varyIfNotSeen}(i, \alpha_n) \end{aligned}$$

where $A = \{\alpha_1, \ldots, \alpha_n\}$ is a finite set of atoms that we suppose ordered in some arbitrary way. Similarly for the set of all agents:

$$\begin{aligned} \mathsf{varyIfNotSeen}(Agt, \alpha) &= \mathtt{JS}\, \alpha? \sqcup (\neg \mathtt{JS}\, \alpha?; (+\alpha \sqcup -\alpha)) \\ \mathsf{varyIfNotSeen}(Agt, A) &= \mathsf{varyIfNotSeen}(Agt, \alpha_1); \cdots ; \\ &\quad\ \mathsf{varyIfNotSeen}(Agt, \alpha_n) \end{aligned}$$

Proposition 4. *Let φ be without $\mathtt{K}_i$ and $\mathtt{CK}$. Then the equivalences*

$$\mathtt{K}_i\varphi \leftrightarrow \mathtt{K}_{\mathsf{varyIfNotSeen}(i, ATM(\varphi))}\varphi$$
$$\mathtt{CK}\varphi \leftrightarrow \mathtt{K}_{\mathsf{varyIfNotSeen}(Agt, ATM(\varphi))}\varphi$$

are valid in the class of all pointed models.

It follows from the above proposition that all modal operators $\mathtt{K}_i$ and $\mathtt{CK}$ can be eliminated from formulas.[2]

Complexity of model checking

Our goal is to design a model checking problem that takes a finite description of an introspective valuation w and a formula φ as an input and says whether $INTR, w \models \varphi$. A first problem is that introspective valuations are always infinite. We are going to represent w in a finite way in order to properly define a model checking decision problem.

From infinite to finite models

We restrict the set of atomic propositions to those relevant for the model checking of φ, noted $RATM(\varphi)$. We obtain $RATM(\varphi)$ by applying Proposition 4, starting with the innermost epistemic operators. The formal definition is by structural induction on φ as follows:

$$\begin{aligned} RATM(\alpha) &= \{\alpha\} \\ RATM(\mathtt{K}_i\varphi) &= RATM(\varphi) \cup \{\mathtt{S}_i\, \alpha : \alpha \in RATM(\varphi)\} \\ RATM(\mathtt{CK}\varphi) &= RATM(\varphi) \cup \{\mathtt{JS}\, \alpha : \alpha \in RATM(\varphi)\} \end{aligned}$$

and homomorphic otherwise. So $RATM(\varphi)$ includes $ATM(\varphi)$. For instance:

$$\begin{aligned} ATM(q \wedge \mathtt{CK}\, \mathtt{K}_i p) &= \{q, p\} \\ RATM(q \wedge \mathtt{CK}\, \mathtt{K}_i p) &= \{q, p, \mathtt{JS}\, p, \mathtt{S}_i\, p, \mathtt{JS}\, \mathtt{S}_i\, p\} \end{aligned}$$

Note that while $RATM(\varphi)$ is finite, its cardinality can be exponential in the length of φ: for example, the cardinality of the set of atoms $RATM(\mathtt{K}_{i_1} \ldots \mathtt{K}_{i_n} p)$ is in 2^n.

Let us define the restriction of valuations and sets thereof to a set of atoms A:

$$\begin{aligned} w|_A &= w \cap A \\ U|_A &= \{w|_A, w \in U\}. \end{aligned}$$

Lemma 1. *Let φ be a formula without epistemic operators ($\mathtt{K}_i$ or $\mathtt{CK}$). Then $RATM(\varphi) = ATM(\varphi)$ and*

$$INTR, w \models \varphi \ \textit{iff} \ INTR|_{RATM(\varphi)}, w|_{RATM(\varphi)} \models \varphi.$$

Proposition 5. *For every formula φ and introspective valuation $w \in INTR$,*

$$INTR, w \models \varphi \ \textit{iff} \ INTR|_{RATM(\varphi)}, w|_{RATM(\varphi)} \models \varphi.$$

Proof. Let φ' be obtained from φ by eliminating the epistemic operators according to the equivalences of Proposition 4. We have:

[2] This uses the rule of replacement of equivalences that preserves validity, cf. Proposition 6.

$INTR, w \models \varphi$;
iff $INTR, w \models \varphi'$ by Proposition 4;
iff $INTR|_{RATM(\varphi)}, w|_{RATM(\varphi)} \models \varphi'$ by Lemma 1;
iff $INTR|_{RATM(\varphi)}, w|_{RATM(\varphi)} \models \varphi$ by Proposition 4. $\square$

The model checking problem

Thanks to Proposition 5, we can restrict w and φ to $RATM(\varphi)$. The model checking problem can then be defined as follows:

- **Input:** a couple $\langle w, \varphi \rangle$ where φ is a formula and w is a finite valuation containing all introspectively valid atoms of $RATM(\varphi)$;
- **Output:** yes if $INTR, w \models \varphi$, no otherwise.

Remark 1. *Note that we do not consider the more general problem of checking a triple $\langle U, w, \varphi \rangle$ where U is a set of valuations. The reason is that the explicit representation of U may require exponential space in the size of w (that may be double-exponential in the length of φ). One might consider representing U by a boolean formula, as done in (Lomuscio, van der Meyden, and Ryan 2000; Su, Sattar, and Luo 2007; van Benthem et al. 2015); however, one cannot represent the set of all introspective valuations $INTR$ in that way.*

Complexity results

The model checking problem was proven to be PSPACE-hard for a logic with less operators (Herzig, Lorini, and Maffre 2015). We now show that the problem is in PSPACE. For that, we adapt the alternating algorithm in (Charrier and Schwarzentruber 2015), originally designed for a variant of a dynamic logic with propositional assignments, public announcements and arbitrary public announcements. Here we consider another, novel variant without arbitrary public announcements but with public assignments. We call this logic DL-PA-PMP (Dynamic Logic of Propositional Assignments with Public and Mental Programs). The syntax of DL-PA-PMP is very similar to the one considered in this paper:

$$\pi ::= \alpha{\leftarrow}\bot \mid \alpha{\leftarrow}\top \mid (\pi;\pi) \mid (\pi \sqcup \pi) \mid \varphi?$$
$$\varphi ::= \alpha \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid [\pi\,!]\varphi \mid \mathtt{K}_\pi\varphi$$

The main difference concerns assignments. The semantics of $\alpha{\leftarrow}\top$ and $\alpha{\leftarrow}\bot$ do not take the introspective consequence relation into account unlike $+\alpha$ and $-\alpha$: we define $w[\alpha{\leftarrow}\top] = w \cup \{\alpha\}$ and $w[\alpha{\leftarrow}\bot] = w \setminus \{\alpha\}$.

Compared to (Charrier and Schwarzentruber 2015), there are two differences. First, general programs $[\pi\,!]\varphi$ are considered, instead of only public announcements. Second, we do not have arbitrary public announcements.

We define the following translation of any formula φ of our logic into DL-PA-PMP:

1. eliminate all epistemic operators from φ, call the resulting formula φ';
2. translate φ' into the DL-PA-PMP formula $tr_{ATM(\varphi')}(\varphi')$ according to the following definition:

$$tr_A(\alpha) = \alpha$$
$$tr_A(\neg\varphi') = \neg tr_A(\varphi')$$
$$tr_A(\varphi_1 \wedge \varphi_2) = tr_A(\varphi_1) \wedge tr_A(\varphi_2)$$
$$tr_A(\mathtt{K}_\pi\varphi) = \mathtt{K}_{tr_A(\pi)} tr_A(\varphi)$$
$$tr_A([\pi\,!]\varphi') = [tr_A(\pi)\,!] tr_A(\varphi')$$
$$tr_A(+\alpha) = \beta_1{\leftarrow}\top; ...; \beta_n{\leftarrow}\top$$
$$tr_A(-\alpha) = \begin{cases} \bot? & \text{if } \alpha INTR \text{ valid} \\ \beta'_1{\leftarrow}\bot; ...; \beta'_m{\leftarrow}\bot & \text{otherwise} \end{cases}$$
$$tr_A(\pi_1;\pi_2) = tr_A(\pi_1); tr_A(\pi_2)$$
$$tr_A(\pi_1 \sqcup \pi_2) = tr_A(\pi_1) \sqcup tr_A(\pi_2)$$
$$tr_A(\varphi?) = tr_A(\varphi)?$$

where

$$\{\beta_1, ..., \beta_n\} = \{\beta \in ATM, \alpha \Rightarrow \beta\} \cap A$$
$$\{\beta'_1, ..., \beta'_m\} = \{\beta \in ATM, \beta \Rightarrow \alpha\} \cap A$$

The size of $tr_{ATM(\varphi')}(\varphi')$ may be exponential in the size of φ, but the size of $\langle w|_{ATM(\varphi')}, tr_{ATM(\varphi')}(\varphi') \rangle$ is polynomial in the size of $\langle w|_{ATM(\varphi)}, \varphi \rangle$. Therefore we have a polynomial reduction from our model checking to the model checking of DL-PA-PMP.

procedure $Mc(w, \varphi)$
  $mc_{yes}([], w, \varphi)$
  accept

Figure 2: The main model checking procedure Mc for DL-PA-PMP.

procedure $mc_{yes}(L, w, \varphi)$
  match φ **with**
    case $\varphi = \alpha$: **if** $\alpha \notin w$ **then reject**
    case $\varphi = \neg\psi$: $mc_{no}(L, w, \psi)$
    case $\varphi = (\psi_1 \wedge \psi_2)$:
      (∀) choose $i \in \{1, 2\}$
      $mc_{yes}(L, w, \psi_i)$
    case $\varphi = \mathtt{K}_\pi\psi$:
      (∀) choose $w' \in W_{all}$
      (∃) $ispath_{no}(L, w, w', \pi)$
        or $survives_{no}(L, w')$
        or $mc_{yes}(L, w', \psi)$
    case $\varphi = [\alpha{\leftarrow}\top!]\psi$:
      $mc_{yes}(L :: (\alpha{\leftarrow}\top), w[\alpha{\leftarrow}\top], \psi)$
    case $\varphi = [\alpha{\leftarrow}\bot!]\psi$:
      $mc_{yes}(L :: (\alpha{\leftarrow}\bot), w[\alpha{\leftarrow}\bot], \psi)$
    case $\varphi = [\pi_1;\pi_2!]\psi$:
      $mc_{yes}(L, w, [\pi_1!][\pi_2!]\psi)$
    case $\varphi = [\pi_1 \sqcup \pi_2!]\psi$:
      $mc_{yes}(L, w, [\pi_1!]\psi \wedge [\pi_2!]\psi)$
    case $\varphi = [\chi?!]\psi$
      (∀) $mc_{yes}(L, w, \chi)$
        and $mc_{yes}(L :: (\chi!), w, \psi)$

Figure 3: The model checking sub-procedure mc_{yes}.

Figures 2, 3, 4 and 5 describe the model checking

procedure $survives_{yes}(L, w)$
  match L **with**
    case $L = []$: do nothing
    case $L = L' :: \varphi!$:
      (∀) $mc_{yes}(L', w, \varphi)$
        and $survives_{yes}(L', w)$
    case $L = L' :: (\alpha{\leftarrow}\top)$:
      (∀) $\alpha \in w$
        and
          (∃) $survives_{yes}(L', w[\alpha{\leftarrow}\bot])$
            or $survives_{yes}(L', w[\alpha{\leftarrow}\top])$
    case $L = L' :: (\alpha{\leftarrow}\bot)$:
      (∀) $\alpha \notin w$
        and
          (∃) $survives_{yes}(L', w[\alpha{\leftarrow}\bot])$
            or $survives_{yes}(L', w[\alpha{\leftarrow}\top])$

Figure 4: The procedure $survives_{yes}$, checking whether a valuation satisfies all announcements made.

procedure $ispath_{yes}(L, w, w', \pi)$
  match π **with**
    case $\pi = \alpha{\leftarrow}\bot$:
      if $w' \neq w \setminus \{\alpha\}$ **then reject**
    case $\pi = \alpha{\leftarrow}\top$:
      if $w' \neq w \cup \{\alpha\}$ **then reject**
    case $\pi = \pi_1; \pi_2$:
      (∃) choose a valuation $v \in W_{all}$
      (∀) $ispath_{yes}(L, w, v, \pi_1)$
        and $ispath_{yes}(L, v, w', \pi_2)$
    case $\pi = \pi_1 \cup \pi_2$:
      (∃) choose $k \in \{1, 2\}$
      $ispath_{yes}(L, w, w', \pi_k)$
    case $\pi = \chi?$:
      (∀) $w = w'$
        and $mc_{yes}(L, w, \chi)$

Figure 5: The path searching procedure $ispath_{yes}$ for mental programs.

procedures for DL-PA-PMP, adapted from (Charrier and Schwarzentruber 2015). In these, the set of all valuations $INTR|_{RATM(\varphi)}$ is fixed in the beginning and is called W_{all}. The procedures are alternating, meaning existential and universal choices are performed. For instance, the instruction '(∀) choose $w' \in U$' succeeds if any choice of w' leads to the accepting state. We also quantify over choices of sub-procedures: for instance, '(∃) $algo_1$ **or** $algo_2$' means that at least one of the calls $algo_1$ or $algo_2$ must succeed.

Figure 2 shows the main model checking procedure for DL-PA-PMP, that calls a sub-procedure mc_{yes} and accepts the input if mc_{yes} does not reject it. We implicitly define the dual of mc_{yes}, called mc_{no}, by replacing **and** by **or**, (∀) by (∃), mc_{yes} by mc_{no} etc. in the pseudo-code of mc_{yes}. We define $survives_{no}$ and $ispath_{no}$ similarly from the code of $survives_{yes}$ and $ispath_{yes}$ respectively.

In Figure 3, the surviving set of valuations is implicitly represented as a list L of announcements and assignments. For this reason, in Figure 2, the set W_{all} is represented by the empty list []. In order to check whether a valuation w' is in U in the K_π case, (that is, the valuation w' survives the announcements and assignments in L), we call $survives_{no}(L, w)$. The sub-procedure $ispath_{yes}(L, w, u, \pi)$, used in the model checking of $\mathtt{K}_\pi\varphi$, checks whether there is a π-path from w with u w.r.t. L. Therefore, in the K_π case, we check that if there is a π-path from w to w' and if w' is in U, then $mc_{yes}(L, w', \varphi)$ must hold.

Note that when choosing between two sub-procedures in '(∃)$algo_1$ **or** $algo_2$' and '(∀)$algo_1$ **and** $algo_2$', either $algo_1$ or $algo_2$ is executed, but not both. Therefore, we can prove that this model checking procedure runs in polynomial time in the size of (w, φ). The model checking of DL-PA-PMP is then in AP. Since AP = PSPACE (Chandra and Stockmeyer 1976), we have shown that our model checking is in PSPACE. Therefore it is PSPACE-complete.

Theorem 1. *The model checking problem is* PSPACE*-complete.*

Proof. As to hardness: the model checking problem was proven to be PSPACE-hard for a logic with less operators (Herzig, Lorini, and Maffre 2015).

As to membership: given an instance $\langle w, \varphi\rangle$, we apply the following algorithm:

- We eliminate all epistemic operators from φ and we obtain φ';
- We compute $tr_{ATM(\varphi')}(\varphi')$;
- We call $Mc(w, tr_{ATM(\varphi')}(\varphi'))$.

This requires a polynomial amount of space in the size of $\langle w, \varphi\rangle$. Therefore, the model checking is in PSPACE. □

More properties

We now discuss a list of valid equivalences.

Equivalences for publicly announced programs

Consider the operator of public program execution $[\pi\,!]$. The following equivalences reduce all programs to either atomic programs or public tests:

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

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

Applying these equivalences exhaustively we obtain programs without sequential and nondeterministic composition. So all programs are either public tests or public assignments.

As to public tests, they can be reduced against all other operators (supposing that the $\mathtt{K}_i$ have been eliminated by Proposition 4):

$$[\chi?\,!]\beta \leftrightarrow \neg\chi \vee \beta\text{, for } \beta \text{ atomic} \qquad (Red_{[?\,!],\beta})$$

$$[\chi?\,!]\neg\varphi \leftrightarrow \neg\chi \vee \neg[\chi?\,!]\varphi \qquad (Red_{[?\,!],\neg})$$

$$[\chi?\,!](\varphi\wedge\varphi') \leftrightarrow [\chi?\,!]\varphi \wedge [\chi?\,!]\varphi' \qquad (Red_{[?\,!],\wedge})$$

$$[\chi?\,!]\mathtt{K}_{\pi'}\varphi \leftrightarrow \neg\chi \vee \mathtt{K}_{\pi'}[\chi?\,!]\varphi \qquad (Red_{[?\,!],\mathtt{K}})$$

Observe that these axioms are exactly the reduction axioms of public announcement logic PAL.

As to positive public assignments, they are deterministic and distribute over the boolean operators:

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

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

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

Similarly, for negative public assignments we have:

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

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

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

However, we do not have a reduction axiom for public assignments followed by mental programs. In particular, the equivalence $[+\alpha\,!]\mathtt{K}_{\pi'}\varphi \leftrightarrow \mathtt{K}_{\pi'}[+\alpha\,!]\varphi$ is invalid. For example, $[+p\,!]\mathtt{K}_{-p}\bot$ is valid while $\mathtt{K}_{-p}[+p\,!]\bot$ is not. Similarly, the equivalence $[-\alpha\,!]\mathtt{K}_{\pi'}\varphi \leftrightarrow \mathtt{K}_{\pi'}[-\alpha\,!]\varphi$ is invalid (unless α is valid in *INTR*).

There also seem to be no reduction axioms for $[\pi\,!][\pi'\,!]\varphi$. Indeed, we would have to apply $(Red_{[;\,!]})$ from the right to the left while we have applied it from the left to the right when reducing programs. An axiomatics without the rule of replacement of equivalents—which is the standard presentation of the axiomatics of PAL and which requires a reduction axiom for $[\pi\,!][\pi'\,!]\varphi$, cf. (Balbiani et al. 2012; Wang and Cao 2013; Wang and Aucher 2013)—seems therefore impossible.

Reduction axioms for mental programs

Consider the mental program operator $\mathtt{K}_\pi$. Reduction axioms for $\sqcup$ and ? are familiar:

$$\mathtt{K}_{\pi\sqcup\pi'}\varphi \leftrightarrow \mathtt{K}_\pi\varphi \wedge \mathtt{K}_{\pi'}\varphi \qquad (Red_{\mathtt{K}_\sqcup})$$

$$\mathtt{K}_{\chi?}\varphi \leftrightarrow \neg\chi \vee \varphi \qquad (Red_{\mathtt{K}_?})$$

So the mental test $\mathtt{K}_{\chi?}$ behaves just as the PDL test. In contrast and as we have seen above, the public test $[\chi?\,!]$ behaves just as the public announcement operator.

However, we do not have reduction axioms for sequential composition; in particular, $\mathtt{K}_{\pi;\pi'}\varphi \leftrightarrow \mathtt{K}_\pi\mathtt{K}_{\pi'}\varphi$ is invalid. This is due to our requirement that mental programs have to terminate in the public information state.

As to positive atomic programs, they distribute over the boolean operators:

$$\mathtt{K}_{+\alpha}\neg\varphi \leftrightarrow \mathtt{K}_{+\alpha}\bot \vee \neg\mathtt{K}_{+\alpha}\varphi \qquad (Red_{\mathtt{K}_+,\neg})$$

$$\mathtt{K}_{+\alpha}(\varphi\wedge\varphi') \leftrightarrow \mathtt{K}_{+\alpha}\varphi \wedge \mathtt{K}_{+\alpha}\varphi' \qquad (Red_{\mathtt{K}_+,\wedge})$$

Similarly, for negative atomic programs:

$$\mathtt{K}_{-\alpha}\neg\varphi \leftrightarrow \begin{cases} \top & \text{if } \alpha \text{ is valid in } INTR \\ \mathtt{K}_{-\alpha}\bot \vee \neg\mathtt{K}_{-\alpha}\varphi & \text{otherwise} \end{cases} \qquad (Red_{\mathtt{K}_-,\neg})$$

$$\mathtt{K}_{-\alpha}(\varphi\wedge\varphi') \leftrightarrow \mathtt{K}_{-\alpha}\varphi \wedge \mathtt{K}_{-\alpha}\varphi' \qquad (Red_{\mathtt{K}_-,\wedge})$$

However, we do not have reduction axioms for mental assignments followed by an atom, i.e., for $\mathtt{K}_{+\alpha}\beta$ and $\mathtt{K}_{-\alpha}\beta$. Again, this is due to our requirement that mental programs have to terminate in the public information state.

Replacement of equivalents

The above equivalences can be applied anywhere in a formula because the inference rule of replacement of equivalents preserves validity. This is due to the fact that the following rules of equivalence for the modal operators $[\pi\,!]$, $\mathtt{K}_\pi$, $\mathtt{K}_i$, $\mathtt{CK}$ all preserve validity:

$$\frac{\varphi\leftrightarrow\psi}{[\pi\,!]\varphi\leftrightarrow[\pi\,!]\psi} \quad \frac{\varphi\leftrightarrow\psi}{\mathtt{K}_\pi\varphi\leftrightarrow\mathtt{K}_\pi\psi} \quad \frac{\varphi\leftrightarrow\psi}{\mathtt{K}_i\varphi\leftrightarrow\mathtt{K}_i\psi} \quad \frac{\varphi\leftrightarrow\psi}{\mathtt{CK}\varphi\leftrightarrow\mathtt{CK}\psi}$$

Proposition 6. *Let φ' be obtained from φ by replacing some occurrence of χ in φ by χ'. Let U be a set of valuations. If $U, w \models \chi \leftrightarrow \chi'$ for every $w \in U$ then $U, w \models \varphi \leftrightarrow \varphi'$ for every $w \in U$.*

Let us terminate this section by pointing out that the equivalences that we have seen do not make up a complete set of reduction axioms. Indeed, while we have equivalences for almost all combinations of operators, we lack reduction axioms for the following three cases.

- Public assignments followed by mental programs: $[+\alpha\,!]\mathtt{K}_{\pi'}\varphi$ and $[-\alpha\,!]\mathtt{K}_{\pi'}\varphi$;
- Sequences of mental programs: $\mathtt{K}_{\pi;\pi'}\varphi$;
- Mental assignments followed by an atom: $\mathtt{K}_{+\alpha}\beta$ and $\mathtt{K}_{-\alpha}\beta$.

We have seen that this is related to our truth condition for mental program operator $\mathtt{K}_\pi$.

We leave a complete axiomatization of the validities of our logic to future work.

Muddy children, proved

In this section we formally prove the statement we made in Example 3 about the muddy children puzzle for the case of two children. Remember that each child sees the other but cannot see herself. This is captured by the following conjunction of visibility atoms:

$$Obs = \neg\mathtt{S}_1\,m_1 \wedge \neg\mathtt{S}_2\,m_2 \wedge \mathtt{S}_1\,m_2 \wedge \mathtt{S}_2\,m_1 \wedge \mathtt{JS}\,\mathtt{S}_1\,m_1 \wedge \mathtt{JS}\,\mathtt{S}_1\,m_2 \wedge \mathtt{JS}\,\mathtt{S}_2\,m_1 \wedge \mathtt{JS}\,\mathtt{S}_2\,m_2.$$

Remember also that initially each child does not know whether she is muddy or not:

$$Ign = \neg\mathtt{K}_1 m_1 \wedge \neg\mathtt{K}_1\neg m_1 \wedge \neg\mathtt{K}_2 m_2 \wedge \neg\mathtt{K}_2\neg m_2.$$

We use the following validity of Public Announcement Logic PAL (which carries over to our logic because all principles of PAL are valid).

Proposition 7. *Let φ and φ' be boolean formulas. Then*

$$(\mathtt{K}_i\varphi \wedge \neg\mathtt{K}_i\varphi' \wedge \neg\mathtt{K}_i\neg\varphi') \to [\varphi\vee\varphi'?\,!](\neg\mathtt{K}_i\varphi' \wedge \neg\mathtt{K}_i\neg\varphi')$$

is valid in pointed epistemic models.

Intuitively, the above proposition says that if agent i knows a fact φ but does not know the fact φ', then publicly announcing that φ or φ' is true does not increase her knowledge about φ'.

The formulas

$$(\mathtt{K}_1 m_2 \wedge \neg\mathtt{K}_1 m_1 \wedge \neg\mathtt{K}_1\neg m_1) \to$$

$$[m_2 \vee m_1?\,!](\neg \mathtt{K}_1 m_1 \wedge \neg \mathtt{K}_1 \neg m_1)$$

and

$$(\mathtt{K}_2 m_1 \wedge \neg \mathtt{K}_2 m_2 \wedge \neg \mathtt{K}_2 \neg m_2) \to [m_1 \vee m_2?\,!](\neg \mathtt{K}_2 m_2 \wedge \neg \mathtt{K}_2 \neg m_2)$$

are instances of the above PAL validity. Observe that $\mathit{Muddy} \wedge \mathit{Obs}$ implies the first conjunct of each of the two antecedents while Ign implies its last two conjuncts. Therefore

$$(\mathit{Muddy} \wedge \mathit{Obs} \wedge \mathit{Ign}) \to [m_1 \vee m_2?\,!]\mathit{Ign}.$$

Moreover,

$$\mathit{Muddy} \to \langle m_1 \vee m_2?!\rangle \top.$$

Putting the last two implications together we obtain:

$$(\mathit{Muddy} \wedge \mathit{Obs} \wedge \mathit{Ign}) \to \langle m_1 \vee m_2?!\rangle \mathit{Ign}.$$

Similarly, we can establish that

$$(\mathit{Muddy} \wedge \mathit{Obs} \wedge \mathit{Ign}) \to \langle m_1 \vee m_2?!\rangle \langle \mathit{Ign}?!\rangle (\mathtt{K}_1 m_1 \wedge \mathtt{K}_2 m_2)$$

is valid.

Conclusion

We have introduced a dynamic epistemic logic where knowledge of agents is deduced from what they see and from what is publicly announced to them. We thereby contribute two main improvements to previous observability-based approaches: we can reason about higher-order knowledge because visibilities are not common knowledge and we can reason about a simple form communication: public announcements. It is the latter feature which allows us to formalize the muddy children puzzle in a natural way.

Beyond public announcements we can reason about publicly executed programs: public announcements are special cases of publicly executed tests. This allows to formalize variants of the muddy children puzzle where the children e.g. clean their forehead (van Ditmarsch, van der Hoek, and Kooi 2005).

One may wonder whether one can identify a validity that is not a validity of the multimodal logic S5. Here is one:

$$\mathtt{K}_1(p \vee q) \to \mathtt{K}_1 p \vee \mathtt{K}_1 q \vee \mathtt{K}_2(p \vee q).$$

This can be explained as follows: for $\mathtt{K}_1(p \vee q)$ to be true, 1 must have learned $p \vee q$ from either observation or communication. In the first case, either she sees p, or she sees q, and therefore either $\mathtt{K}_1 p$ or $\mathtt{K}_1 q$ is true. In the second case, $p \vee q$ must have been announced and thus $\mathtt{K}_2(p \vee q)$ is true.

While we proved that the model checking problem is PSPACE-complete, it remains to prove the complexity of the satisfiability problem. Other future works would be to generalize the common knowledge operator to an arbitrary subset of agents or add the Kleene star to the language of programs. It also remains to find an axiomatization.

Beyond public announcement logic, we would like to investigate to which extent one can capture so-called action models of dynamic epistemic logics (Baltag and Moss 2004; van Ditmarsch, van der Hoek, and Kooi 2007). We have shown in (Herzig, Lorini, and Maffre 2015) that we can also model at least some forms of private announcements thanks to our visibility atoms. For example, that φ is true after the private announcement of p to i is captured by the formula $p \to [+\mathtt{S}_i\, p\,!]\varphi$. We have furthermore modeled private announcements of literals, of conjunctions of literals and of knowledge of literals: for example, φ is true after the private announcement of $\mathtt{K}_i p$ to j if and only if $\mathtt{K}_i p \to [+\mathtt{S}_j\, \mathtt{S}_i\, p; +\mathtt{S}_j\, p\,!]\varphi$ is true.[3] We have shown in (Herzig and Maffre 2015) that this allows to formalize the gossip problem: there are n friends, and each i knows a fact s_i not known to anybody else. They can call each other and during a call they exchange all their knowledge. The problem is to find a minimal number of calls allowing to reach a state where all secrets are shared knowledge: each agent knows every secret. We have shown that a generalized version of the problem where the goal is to achieve shared knowledge up to a fixed depth can also be formalized in our logic. Other variants can be captured as well, such as when there are agents that cannot phone to each other, when the goal is that some agents remain ignorant about some secrets, and when secrets (that might be better thought of as passwords in that case) can be changed by their owner. One might imagine to find combinations of gossip problems and muddy children puzzles…

References

Alechina, N.; Logan, B.; and Whitsey, M. 2004. A complete and decidable logic for resource-bounded agents. In *Proceedings of AAMAS 2014*, 606–613. IEEE Computer Society.

Balbiani, P.; van Ditmarsch, H.; Herzig, A.; and de Lima, T. 2012. Some truths are best left unsaid. In Bolander, T.; Braüner, T.; Ghilardi, S.; and Moss, L. S., eds., *Advances in Modal Logic 9, papers from the ninth conference on "Advances in Modal Logic," held in Copenhagen, Denmark, 22-25 August 2012*, 36–54. College Publications.

Balbiani, P.; Fernandez-Duque, D.; and Lorini, E. 2016. A logical theory of belief dynamics for resource-bounded agents. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. International Foundation for Autonomous Agents and Multiagent Systems.

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

Balbiani, P.; Herzig, A.; and Troquard, N. 2013. 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*, 143–152.

Baltag, A., and Moss, L. S. 2004. Logics for epistemic programs. *Synthese* 139(2):165–224.

[3]Furthermore, φ is true after the private announcement that i "knows whether" p to j if and only $\mathtt{S}_i\, p \to [+\mathtt{S}_j\, \mathtt{S}_i\, p\,!]\varphi$ is true. Note that we indeed communicate less information to j when announcing her $\mathtt{S}_i\, p$ than announcing her $\mathtt{K}_i p$.

Castelfranchi, C. 1994. Guarantees for autonomy in cognitive agent architecture. In Wooldridge, M., and Jennings, N. R., eds., *Intelligent Agents, Proceedings of the 1994 ECAI Workshop on Agent Theories, Architectures and Languages*, volume 890 of *LNCS/LNAI*, 56–70. Springer-Verlag.

Chandra, A. K., and Stockmeyer, L. J. 1976. Alternation. In *Proc. of FOCS'76*, 98–108.

Charrier, T., and Schwarzentruber, F. 2015. Arbitrary public announcement logic with mental programs. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 1471–1479. International Foundation for Autonomous Agents and Multiagent Systems.

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

Gasquet, O.; Goranko, V.; and Schwarzentruber, F. 2014. Big brother logic: logical modeling and reasoning about agents equipped with surveillance cameras in the plane. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014*, 325–332.

Grant, J.; Kraus, S.; and Perlis, D. 2000. A logic for characterizing multiple bounded agents. *Journal of Autonomous Agents and Multi-Agent Systems* 3(4):351–387.

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

Herzig, A., and Maffre, F. 2015. How to share knowledge by gossiping. In Julian Inglada, V. J., and Vouros, G., eds., *3rd International Conference on Agreement Technologies, Athens, 17.12.2015-18.12.2015*, LNCS/LNAI. Springer-Verlag.

Herzig, A.; Lorini, E.; Troquard, N.; and Moisan, F. 2011. A dynamic logic of normative systems. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 228–233.

Herzig, A.; Lorini, E.; and Maffre, F. 2015. A poor man's epistemic logic based on propositional assignment and higher-order observation. In *Proceedings of the 5th International Conference on Logic, Rationality, and Interaction (LORI)*. Springer Berlin Heidelberg.

Lehmann, D. J. 1984. Knowledge, common knowledge and related puzzles (extended summary). In *PODC*, 62–67.

Lomuscio, A.; van der Meyden, R.; and Ryan, M. 2000. Knowledge in multiagent systems: initial configurations and broadcast. *ACM Trans. Comput. Log.* 1(2):247–284.

Su, K.; Sattar, A.; and Luo, X. 2007. Model checking temporal logics of knowledge via OBDDs. *Comput. J.* 50(4):403–420.

van Benthem, J.; van Eijck, J.; Gattinger, M.; and Su, K. 2015. Symbolic model checking for Dynamic Epistemic Logic. In van der Hoek, W.; Holliday, W. H.; and Wang, W.-f., eds., *Proceedings of the 5th International Conference on Logic, Rationality and Interaction*, 366–378. Springer-Verlag.

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

van der Hoek, W.; Troquard, N.; and Wooldridge, M. 2011. Knowledge and control. In Sonenberg, L.; Stone, P.; Tumer, K.; and Yolum, P., eds., *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, 719–726. IFAAMAS.

van Ditmarsch, H. P.; van der Hoek, W.; and Kooi, B. 2005. Dynamic epistemic logic with assignment. In *Proc. AAMAS'05*, 141–148.

van Ditmarsch, H.; van der Hoek, W.; and Kooi, B. 2007. *Dynamic Epistemic Logic*. Springer Publishing Company, Incorporated, 1st edition.

van Linder, B.; Hoek, W. V. D.; and Meyer, J.-J. C. 1997. Seeing is believing: and so are hearing and jumping. *Journal of Logic, Language and Information* 6(1):33–61.

Wang, Y., and Aucher, G. 2013. An alternative axiomatization of DEL and its applications. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*.

Wang, Y., and Cao, Q. 2013. On axiomatizations of public announcement logic. *Synthese* 190(Supplement-1):103–134.