Knowledge about lights along a line

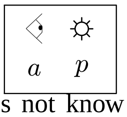
François SCHWARZENTRUBER

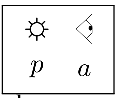
***Abstract*—In this article, we are going to talk about spatial situations. Every agent (human, camera etc.) and every proposition (lamp, object, etc.) are located in the space (here a line) and we express properties over a situation using standard epistemic logic language possibly extended with public announcements. We study links between validities of this *geometrical* version of epistemic logic and the standard one. We also investigate complexities of model checking and satisfiability.**

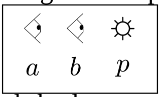
Keywords: Multi-agent system. Epistemic logic. Spatial reasoning. Public announcements. Pedagogical tool. Complexity theory. Polynomial hierarchy.

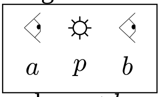
I. INTRODUCTION

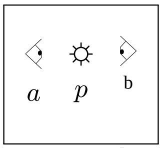
Many authors in logic and in Artificial Intelligence [5] developed epistemic logic and studied mathematical properties of it. Epistemic logic is theoretical and may be difficult to explain to students. This is the reason why in this article we are going to study a concrete example of multi-agent system. Let us take a line. We are going to put lamps and agents on this line as shown in the Figure 1. Now the question is "what do agents know about lamps and knowledge of other agents about lamps"? This system has been implemented as a pedagogical tool in order to illustrate any epistemic logic course. Indeed, students can easily understand some epistemic logic on concrete examples:

- Agent a sees the lamp p on, so he knows p; 

- Agent a does not see the lamp p, so he does not know whether p or $\neg p$; 

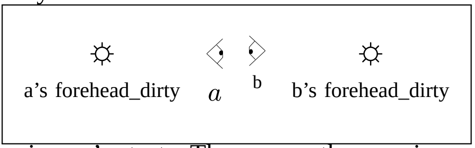
- Agent a sees another agent b seeing the lamp p on, so agent a knows agent b knows p; 

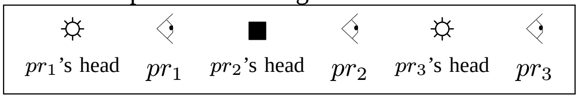
- Agent a sees another agent b, and the lamp p, but agent a sees that agent b is not looking in the direction of the lamp p, so agent a knows agent b does not know whether p or $\neg p$; 

- Agent a and agent b are looking at each other and there is the lamp p between them, so there is common knowledge that p is true. This kind of situations have already been considered in [8]. 

This approach can be compared to the approach in [2] for first order logic. In [2], you put objects like cube, pyramids and you can then write formulas in first order logic to check properties over those objects. Here the approach is similar: you put agents and lamps and then you can write formulas in epistemic logic to check whether properties over those agents and lamps are true.

Generally speaking many examples of epistemic situation mix time and space. The link between time and knowledge (perfect recall etc.) has been studied and you can find a survey in [3]. There exists also some work linking space and knowledge like in [10]: they provide a logic with a spatial modal operator dealing with topology and an epistemic modal operator. Here, our approach is different: we want to deal with a spatially grounded epistemic logic. We are not going to provide operators in the language to deal with space but only provide an epistemic operator for each agent in the language. The semantics will then directly rely on the geometrical properties of a line. We would like to describe a situation but directly by the graphical and natural representation of the system and not with a Kripke structure. We can formalize well known toy examples as Russian cards [13], Muddy children ([4], [11], [3]) or the prisoner's test. For instance:

- The Muddy children. The spatial configuration is the following: two children are looking at each other. One knows the other's forehead is dirty. But one does not know he is dirty. 

- The prisoner's test. There are three prisoners on a line. Each prisoner must guess the color of his head. 

Another motivation would be video surveillance. Propositions are objects we have to take care of. Agents are camera. We then can specify the video surveillance with epistemic logic formulas. Another possible application may be robots' space and knowledge reasoning because robots evolve in our spatial world. A last application could be video games. In many role playing games or strategy games, players or non-playing characters can have knowledge about the virtual world. The behaviours of a non-playing character can then be described by the game designer using a knowledge based programming language. For instance, the designer can specify that the guardian of the castle gets crazy if he knows that the door of the castle is open. A preliminary work about formalizing the video game Thief has been done in [7].

A piece of software is available on the Web Site http://www.irit.fr/~Francois.Schwarzentruber/agentsandlamps/. It provides a model-checker: you specify the graphical situation and a formula written using epistemic modal operators and/or public announcements operators. In this article:

- We are going to present the semantics of the geometric version of epistemic logic in section II;
- We are going to deal with the model checking and satisfiability problems' complexities in section III;
- In section IV, we will add public announcements to our language to model examples like Muddy children;
- In section V, we are going to present the current implementation.

II. Semantics

We are going to define a new logic based on the same language than the epistemic logic $S5_n$ [5]. $S5_n$ is the logic of frames where relations are equivalence relations. Here we are defining a logic where the semantics is based on a geometric point of view.

A. Language

Our logic is based on the same language as $S5_n$'s one. Let us recall the language of the epistemic logic $S5_n$ [5].

Definition 1 (language):
Let ATM be a countable set of atomic propositions. Let AGT be a countable set of agents. The language $\mathcal{L}_{AGT}$ is defined by the following BNF:

$$\varphi ::= \top \mid p \mid \varphi \wedge \varphi \mid \neg\varphi \mid K_a\psi$$

where $p \in ATM$ and $a \in AGT$.

As usual, $\varphi \vee \psi =^{def} \neg(\neg\varphi \wedge \neg\psi)$. $\hat{K}_a\psi =^{def} \neg K_a\neg\psi$.

Notice that we can only deal with knowledge (operator K_a) and states of lamps (proposition p is true means that the lamp called p is on) in the language. One may expect to deal also with position of lamps, position of agents or maybe spatial topologic operator like in [10] etc. This may be very interesting, especially in all applications cited in the introduction. For instance, we can not express a sentence like "the guardian knows that the beetle is *near* the old man." but we can say "The guardian knows that the beetle knows the old man's hat is red." ($K_{guardian}K_{beetle}old\_man\_red$) Here we have preferred to keep the language of classical epistemic logic for two reasons:

- a pedagogical tool for understanding epistemic logic should be simple and should have a simple syntax;
- to focus on complexity results with the simple expressivity as $S5_n$.

B. Definitions

The semantics is not defined with a class of models but directly from what a concrete situation is. From this, we will obtain a spatially grounded epistemic logic. A *world* is situation where all agents have a *location* (*position* and *direction* where they look), all *lamps* (atomic propositions) have a *location* and a *state* (on or off). Formally:

Definition 2 (world):
A *world* w is a tuple $\langle p_{AGT}, d_{AGT}, p_{ATM}, \pi\rangle$ where:

- $p_{AGT} : AGT \to \mathbb{R}$;
- $d_{AGT} : AGT \to \{-1, +1\}$;
- $p_{ATM} : ATM \to \mathbb{R}$;
- $\pi : ATM \to \{\bot, \top\}$.

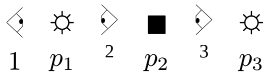


Fig. 1. Example of a world

The set of all worlds is noted W.

In a world $\langle p_{AGT}, d_{AGT}, p_{ATM}, \pi\rangle$, $p_{AGT}(a)$ denotes the position of agent a. $d_{AGT}(a)$ denotes the direction where the agent a looks: if $d_{AGT}(a) = +1$, the agent a will look on the right and if $d_{AGT}(a) = -1$, he will look on the left. $p_{ATM}(p)$ denotes the position of the lamp saying whether p is true or not. $\pi(p) = \top$ iff the lamp "p" is on. $\pi(p) = \bot$ means that the lamp "p" is off.

We have defined a world in the more close to the reality manner: that is to say using the real numbers. We could also consider locations of agents and lamps as a total preorder over $ATM \cup AGT$. Considering a total preorder is discussed at the end of this section and total preorder is used in Section III. Here we prefer to use the Definition 8 whose advantage is that it can be easily generalized to dimension $n \geq 2$: you just have to replace $\mathbb{R}$ by $\mathbb{R}^n$ and to adapt the notion of direction. In dimension 2 or more, total preorders can no longer be used.

We can also discuss the Definition 8 by the way propositions are treated. Here, a proposition p is associated to a point $p_{ATM}(p)$. This seems to be the simple way to define the semantics. But be aware that in some cases this is a limitation:

- Maybe a proposition p can be associated to a set of points. For instance, if you are at home, you can know it rains either by looking towards the window of the left L or the window of the right R. Hence, here the proposition $rain$ may be associated to the set of points $\{L, R\}$;
- Maybe you want that a lamp is associated not to a proposition but more generally to a formula. For instance, when you know that the alarm system located on point P is on, you in fact know that either there is an oil problem or overheating. Hence, here the point is associated to the formula $oil\_problem \vee overheating$.

Here we stay with the simple definition for two reasons:

- it is easier for a pedagogical tool to have a simple and clear semantics;
- it is easier for us to begin study a simple case.

Definition 3 (cone):
Let us consider a world $w = \langle p_{AGT}, d_{AGT}, p_{ATM}, \pi\rangle$. We note $cone(a)$ the set $\{p_{AGT}(a) + \lambda.d_{AGT}(a) \mid \lambda \in \mathbb{R}^+\}$.

$cone(a)$ denotes all the set of points the agent a sees.

Example 1: The Figure 1 gives us an example of a world w. We have:

- $p_{AGT}(1) = 0; p_{AGT}(2) = 2; p_{AGT}(3) = 4;$
- $d_{AGT}(1) = +1; d_{AGT}(2) = -1; d_{AGT}(3) = -1;$
- $p_{ATM}(p_1) = 1; p_{ATM}(p_2) = 3; p_{ATM}(p_3) = 5;$
- $\pi(p_1) = \top; \pi(p_2) = \bot; \pi(p_3) = \top;$
- $cone(1) = [0, +\infty[;$
- $cone(2) =]-\infty, 2];$

- $cone(3) =]-\infty, 4]$.

Now we are going to define the epistemic relation over worlds. $wR_a u$ means that agent a can not distinguish w from u. In other words, $wR_a u$ iff agent a sees the same things in w and u. Formally:

Definition 4 (epistemic relation):
Let $a \in \mathit{AGT}$. We define the relation R_a over worlds: $\langle p_{\mathit{AGT}}, d_{\mathit{AGT}}, p_{ATM}, \pi\rangle R_a \langle p'_{\mathit{AGT}}, d'_{\mathit{AGT}}, p'_{ATM}, \pi'\rangle$ iff for all $b \in \mathit{AGT}$, for all $p \in ATM$,

- if $p_{\mathit{AGT}}(b) \in cone(a)$ then
  $p_{\mathit{AGT}}(b) = p'_{\mathit{AGT}}(b)$ and $d_{\mathit{AGT}}(b) = d'_{\mathit{AGT}}(b)$;
- if $p_{\mathit{AGT}}(b) \notin cone(a)$ then $p'_{\mathit{AGT}}(b) \notin cone(a)$;
- if $p_{ATM}(p) \in cone(a)$ then
  $p_{ATM}(p) = p'_{ATM}(p)$ and $\pi(p) = \pi'(p)$
- if $p_{ATM}(p) \notin cone(a)$ then $p'_{ATM}(p) \notin cone(a)$.

Briefly, suppose that $wR_a u$. If agent a see the agent b in the world w, then he will also see agent b in world u and agent b will have the same location (position and direction). If agent a does not see agent b in the world w, then he also does not see agent b in u. If agent a see the lamp p in the world w, then he will also see the lamp p in world u. The lamp will have the same position and state both in w and u. If agent a does not see the lamp p in w, then he will also not see the lamp p in u.

Until now, we have finally defined a model $\mathcal{M} = \langle W, (R_a)_{a \in \mathit{AGT}}, \nu\rangle$ where ν maps each world $w \in W$ to π_w. From now, the truth conditions is standard:

Definition 5 (truth conditions):
Let $w \in W$. We define $w \models \varphi$ by induction:

- $w \models \top$;
- $w \models p$ iff $\pi(p) = \top$
- $w \models \varphi \wedge \psi$ iff $w \models \varphi$ and $w \models \psi$;
- $w \models \neg\varphi$ iff $w \not\models \varphi$;
- $w \models K_a\psi$ iff for all w', $wR_a w'$ implies $w' \models \psi$.

C. Comparison with epistemic logic

Now we are going to compare the epistemic logic $S5_n$ and the set of validities we obtain with the truth conditions of Definition 12. First we give the definition of validities.

Definition 6 (set of validities):
We denote the set of all validities by $L^{\text{☼}1D}$, that is to say, $L^{\text{☼}1D} = \{\varphi \in \mathcal{L}_{\mathit{AGT}} \mid \forall w \in W, w \models \varphi\}$.

In "$L^{\text{☼}1D}$", "$1D$" stands for "one dimension" (a line). Now, we can see that our set $L^{\text{☼}1D}$ contains all validities of $S5_n$.

Proposition 1: $S5_n \subseteq L^{\text{☼}1D}$.

Proof: We prove that for all $a \in \mathit{AGT}$, the relation R_a is an equivalence relation. Hence, the model $\mathcal{M}$ is a model of the logic $S5_n$ and satisfies validities of $S5_n$. We have to prove reflexivity, symmetry and transitivity. Let us just begin to prove transitivity. Suppose we have: $\langle p_{\mathit{AGT}}, d_{\mathit{AGT}}, p_{ATM}, \pi\rangle R_a \langle p'_{\mathit{AGT}}, d'_{\mathit{AGT}}, p'_{ATM}, \pi'\rangle$ and $\langle p'_{\mathit{AGT}}, d'_{\mathit{AGT}}, p'_{ATM}, \pi'\rangle R_a \langle p''_{\mathit{AGT}}, d''_{\mathit{AGT}}, p''_{ATM}, \pi''\rangle$. Let us prove that $\langle p_{\mathit{AGT}}, d_{\mathit{AGT}}, p_{ATM}, \pi\rangle R_a \langle p''_{\mathit{AGT}}, d''_{\mathit{AGT}}, p''_{ATM}, \pi''\rangle$.

First we have $p_{\mathit{AGT}}(a) \in cone(a)$. So $p_{\mathit{AGT}}(a) = p'_{\mathit{AGT}}(a) = p''_{\mathit{AGT}}(a)$ and $d_{\mathit{AGT}}(a) = d'_{\mathit{AGT}}(a) = d''_{\mathit{AGT}}(a)$. In other words, $cone(a) = cone'(a) = cone''(a)$.

From now on, if $p_{\mathit{AGT}}(b) \in cone(a)$, then $p_{\mathit{AGT}}(b) = p'_{\mathit{AGT}}(b)$ and $d_{\mathit{AGT}}(b) = d'_{\mathit{AGT}}(b)$. But, we have effectivelly, $p_{\mathit{AGT}}(b) = p'_{\mathit{AGT}}(b)$ and $cone(a) = cone'(a)$. So $p'_{\mathit{AGT}}(b) \in cone'(a)$. So $p'_{\mathit{AGT}}(b) = p''_{\mathit{AGT}}(b)$ and $d'_{\mathit{AGT}}(b) = d''_{\mathit{AGT}}(b)$. Finally, $p_{\mathit{AGT}}(b) = p''_{\mathit{AGT}}(b)$ and $d_{\mathit{AGT}}(b) = d''_{\mathit{AGT}}(b)$. The other cases are treated in the same manner. ■

The semantics of $K_a p$ in $L^{\text{☼}1D}$ corresponds to the fact that the agent a sees the light p and the light p is on. More generally, $K_a\psi$ means that the agent a has the proof that ψ. That is why we have those validities in $L^{\text{☼}1D}$:

Proposition 2: Let $p, q \in ATM$.
$\models_{L^{\text{☼}1D}} K_1(p \vee q) \rightarrow K_1 p \vee K_1 q$.
$\models_{L^{\text{☼}1D}} K_1(\neg p \vee \neg q) \rightarrow K_1\neg p \vee K_1\neg q$.
If $p \neq q$, $\models_{L^{\text{☼}1D}} K_1(p \vee \neg q) \rightarrow K_1 p \vee K_1\neg q$

Proof: Let us prove $\models_{L^{\text{☼}1D}} K_1(p \vee q) \rightarrow K_1 p \vee K_1 q$. Let $w = \langle p_{\mathit{AGT}}, d_{\mathit{AGT}}, p_{ATM}, \pi\rangle$ be a world such that $w \models K_1(p \vee q)$. We are going to prove that either $w \models K_1 p$ or $w \models K_1 q$. We have $p_{ATM}(p) \in cone(1)$ or $p_{ATM}(q) \in cone(1)$. Indeed, if we suppose the contrary, that is to say $p_{ATM}(p) \notin cone(1)$ and $p_{ATM}(q) \notin cone(1)$, there exists a world $u = \langle p_{\mathit{AGT}}, d_{\mathit{AGT}}, p_{ATM}, \pi'\rangle$ such that $\pi'(p) = \bot$ and $\pi'(q) = \bot$ and $wR_1 u$. Hence, $w \not\models K_1(p \vee q)$. Contradiction. So $p_{ATM}(p) \in cone(1)$ or $p_{ATM}(q) \in cone(1)$. For instance, $p_{ATM}(p) \in cone(1)$. And for all $u \in R_1(w)$, $\pi_u(p) = \top$. So $w \models K_1 p$. The other cases are treated in the same manner. ■

Informally, $K_1(p \vee q)$ means that agent 1 has a proof that $p \vee q$. In other words, either he sees p on, or he sees q on. Hence, either $K_1 p$ or $K_1 q$. Nevertheless, $K_1(\varphi \vee \psi) \rightarrow K_1\varphi \vee K_1\psi$ is not valid in $L^{\text{☼}1D}$.

Notice that there are crucial differences between $S5_n$ and $L^{\text{☼}1D}$:

- $S5_n$ is defined as the logic of a *class* of frames and has the property of uniform substitution. If $\models_{S5_n} \varphi[p]$, we have $\models_{S5_n} \varphi[\psi/p]$ for every formula $\psi \in \mathcal{L}_{\mathit{AGT}}$;
- On the contrary (see Definition 12), $L^{\text{☼}1D}$ is defined as the set of formulas valid on *one* model: the model $\mathcal{M}$. As the definition of R_a (Definition 4) depends on worlds, and especially on valuations, it is not surprising that $L^{\text{☼}1D}$ does not have the property of uniform substitution. A just *one* model semantics may seem a poor pedagogical application. But, the model $\mathcal{M}$ is big (if *AGT* and ATM are finite, the size of $\mathcal{M}$ is exponential in *card*($ATM \cup \mathit{AGT}$). In fact, you can imagine the model $\mathcal{M}$ to be a kind of canonical model. The model $\mathcal{M}$ is made up with many connected components. For instance, Figure 2 and 8 show two connected components of the model $\mathcal{M}$.

Now, here is a Proposition showing that we can have common knowledge only when $K_1K_2p \wedge K_2K_1p$.

Proposition 3: We have:
$\models_{L^{\text{☼}1D}} K_1K_2p \wedge K_2K_1p \rightarrow K_1K_2K_1\ldots K_2\ldots p$ where "$K_1K_2K_1\ldots K_2\ldots$" denotes any finite sequence of K_1 and K_2.

Proof: Let $w = \langle p_{\mathit{AGT}}, d_{\mathit{AGT}}, p_{ATM}, \pi\rangle$ be world such that $w \models K_1K_2p \wedge K_2K_1p$. We want to prove that $w \models K_1K_2K_1\ldots K_2\ldots p$. We are going to prove that:

- $p_{\mathit{AGT}}(2) \in cone(1)$;

- $p_{AGT}(1) \in cone(2)$;
- $p_{ATM}(p) \in cone(1)$;
- $p_{ATM}(p) \in cone(2)$.

Let us prove $p_{ATM}(p) \in cone(1)$ by contradiction. Suppose that $p_{ATM}(p) \notin cone(1)$. Thus there exists a world $w' = \langle p'_{AGT}, d'_{AGT}, p'_{ATM}, \pi' \rangle$ such that wR_1w' and $\pi'(p) = \bot$.

We have $w' \not\models p$ so $w' \not\models K_2p$. So $w \not\models K_1K_2p$. Contradiction.

Same proof for $p_{ATM}(p) \in cone(2)$.

Let us prove that $p_{AGT}(2) \in cone(1)$ by contradiction. Suppose that $p_{AGT}(2) \notin cone(1)$. Thus there exists a world $w' = \langle p'_{AGT}, d'_{AGT}, p'_{ATM}, \pi' \rangle$ such that $w\ R_1w'$ and $d'_{AGT}(2)$ is such that $p_{ATM}(p) \notin cone'(2)$. Thus, there exists a world $w'' = \langle p''_{AGT}, d''_{AGT}, p''_{ATM}, \pi'' \rangle$ such that $w'R_2w''$ and $\pi''(p) = \bot$. So $w' \not\models K_2p$. Hence $w \not\models K_1K_2p$. Contradiction.

Same proof for $p_{AGT}(1) \in cone(2)$.

Now we can prove by induction on n that for $n \in \mathbb{N}$, for all $u \in (R_1 \circ R_2)^n(w)$, we have:

- $p_{AGT}(2) \in cone(1)$;
- $p_{AGT}(1) \in cone(2)$;
- $p_{ATM}(p) \in cone(1)$;
- $p_{ATM}(p) \in cone(2)$.
- $\pi(p) = \top$.

Hence $w \models K_1K_2K_1 \ldots K_2 \ldots p$. ■

The validity $K_1K_2p \wedge K_2K_1p \rightarrow K_1K_2K_1 \ldots K_2 \ldots p$ expresses that if $K_1K_2p \wedge K_2K_1p$ then the state of the lamp p is the topic of a *mutual social perception*, studied in [8].

Corollary 1: If $n \geq 2$ or *card*$(ATM) \geq 2$, $S5_n \subsetneq L^{\text{☼}1D}$.

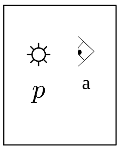
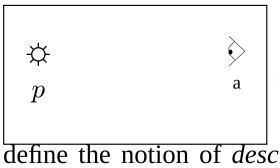
Proof: The formula $K_1(p \vee q) \rightarrow K_1p \vee K_1q$ and $K_1K_2p \wedge K_2K_1p \rightarrow K_1K_2K_1p$ are in $L^{\text{☼}1D}$but are not valid in $S5_n$. ■

More surprising is the fact that common knowledge is not guaranteed by $K_1K_2\varphi \wedge K_2K_1\varphi$ for all φ. More precisely, $K_1K_2\varphi \wedge K_2K_1\varphi \rightarrow K_1K_2K_1\varphi$ is not $L^{\text{☼}1D}$-valid for all φ. Look at the model of the Figure 2: agent 1 = agent in blue. agent 2 = agent in red. Consider the world on the bottom on the right. Let us call it w. We have $w \models K_1K_2\neg K_2p \wedge K_2K_1\neg K_2p$. But, we have $w \not\models K_1K_2K_1\neg K_2p$. Indeed there exists w' such that $wR_1 \circ R_2 \circ R_1w'$ such that $w' \models K_2p$.

Nevertheless, there are other formulas where it remains true. For instance, we have $\models_{L^{\text{☼}1D}} K_1K_2K_3p \wedge K_2K_1K_3p \rightarrow K_1K_2K_1 \ldots K_2 \ldots K_3p$.

Question 1: What about $K_1K_2\varphi \wedge K_2K_1\varphi \rightarrow K_1K_2K_1\varphi$ if φ do not contain agent 1 or 2? Do we have a characterisation or exhibit an interesting set of formulas φ such that $K_1K_2\varphi \wedge K_2K_1\varphi \rightarrow K_1K_2K_1\varphi$ holds?

D. A compact representation

Last but not the least, you can remark that if we want to deal with model-checking, satisfiability problem and other algorithmic problems, we need a compact representation that an algorithm can manipulate. Worlds are difficult to manipulate: in particular, it is unadapted that $R_a(w)$ is infinite given a agent a and a world w. According to the Definition 8, the set W is infinite. Nevertheless, the semantics do not depend on positions of lamps and agents but only on how they are ordered on the line. For instance,  and  stands for the same world. We can define the notion of *description of a world* w: it is simply a total preorder over all propositions and agents appearing in a formula, plus d_{AGT} and π. Notice that we can do this because the space is a line. If our space were $\mathbb{R}^n$ $(n \geq 2)$, the notion of total preorder would unfortunately not be suited anymore.

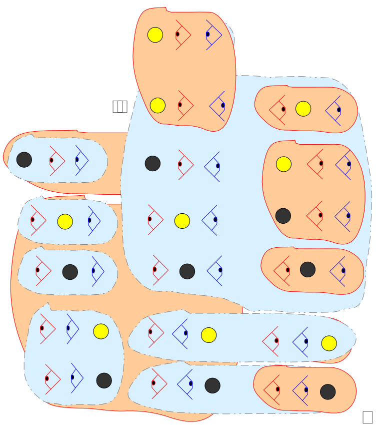


Fig. 2. Some worlds of the model $\mathcal{M}$

Definition 7 (description of a world): A *description of a world* w is a tuple $\langle <, d_{AGT}, \pi \rangle$ where:

- $\leq$ is a total preorder over $AGT \cup ATM$;
- $d_{AGT} : AGT \rightarrow \{-1, +1\}$;
- $\pi : ATM \rightarrow \{\bot, \top\}$.

We can also define the epistemic relation between two description of a world w:

Definition 8 (epistemic relation): Let $a \in AGT$. We define the *epistemic relation* R_a on the set of descriptions of worlds by wR_av iff:

- if $d_{AGT}(a) = +1$,
  - for all $x \in AGT \cup ATM$, $(x \leq_w a$ iff $x \leq_v a)$;
  - for all $x, y \in AGT \cup ATM$ such that $a \leq_w x$ and $a \leq_w y$, we have $(x \leq_w y$ iff $x \leq_v y)$;
  - for all $x \in AGT$, $a \leq_w x$ implies $d_{AGT_w}(x) = d_{AGT_v}(y)$;
  - for all $x \in ATM$, $a \leq_w x$ implies $\pi_w(x) = \pi_v(y)$.
- if $d_{AGT}(a) = -1$,
  - for all $x \in AGT \cup ATM$, $(x \geq_w a$ iff $x \geq_v a)$;
  - for all $x, y \in AGT \cup ATM$ such that $a \geq_w x$ and $a \geq_w y$, we have $(x \geq_w y$ iff $x \geq_v y)$;
  - for all $x \in AGT$, $a \geq_w x$ implies $d_{AGT_w}(x) = d_{AGT_v}(y)$;
  - for all $x \in ATM$, $a \geq_w x$ implies $\pi_w(x) = \pi_v(y)$.

In the same way, we can define an epistemic model. We can define truth conditions of a formula φ in $\mathcal{L}_{AGT}$ over the set of descriptions of worlds, using the epistemic relation. We can prove that we obtain the same validities.

Definition 9 (extracting description of world from a world):

Given a world w, we define the description of world $d(w)$ by:

- for all $x, y \in AGT \cup ATM$, $x \leq_{d(w)} y$ iff $p(x) \leq_{\mathbb{R}} p(y)$ where $p(x)$ stands for $p_{AGT}(x)$ if $x \in AGT$ or $p_{ATM}(x)$ if $x \in ATM$;
- $d_{AGTw} = d_{AGTd(w)}$;
- $\pi_w = \pi_v$.

Proposition 4: For all $w \in W$, for all $\varphi \in \mathcal{L}_{AGT}$, $w \models \varphi$ iff $d(w) \models \varphi$.

Proof: By induction on φ. ■

In the case of one dimension, we simply rewrite mapping from ATM or AGT to real numbers into a total preorder over $ATM \cup AGT$. In the case of two or more dimensions, it is an open problem how to represent a world in a compact way.

III. Model-checking and satisfiability problem

For definitions for complexity class and for more details about the problem QSAT (quantified boolean formulas satisfiability problem), the reader may refer to [9].

A. Definitions

Now we are going to recall the classical problem of model-checking and satisfiability. The problem of model-checking consists on testing if a given formula φ is true in a given world w. Satisfiability problem consists to test if there exists a world w in which a given formula φ is true.

Definition 10 (model-checking of $L^{✡1D}{}_{AGT,ATM}$):
Let AGT be a set of agents and ATM a set of atoms. We call *model-checking of $L^{✡1D}{}_{AGT,ATM}$* problem the following problem:

- Input: a formula $\varphi \in \mathcal{L}_{AGT}$, a description of a world w where only atoms and agents occurring in φ are given;
- Output: Yes iff we have $w \models_{L^{✡1D}} \varphi$. No, otherwise.

In the previous Definition, we give a description of a world w that is to say a total preorder over all agents and propositions occurring in φ where we say for each agent if he is look on the left or on the right and for each proposition if it is true or not. We do not care about propositions or agents not in the formula φ. The description of w is then *finite*.

Definition 11 ($L^{✡1D}{}_{AGT}$-satisfiability problem):
Let AGT be a set of agents. We call $L^{✡1D}{}_{AGT}$-satisfiability problem the following problem:

- Input: a formula $\varphi \in \mathcal{L}_{AGT}$;
- Output: Yes iff there exists a world w such that $w \models_{L^{✡1D}} \varphi$. No, otherwise.

```
function $check(w, \varphi)$
    match ($\varphi$)
        $\top$:
            return $\top$;
        $p \in ATM$:
            return $\top$ if $p$ is true in $w$;
            return $\bot$ if $p$ is false in $w$;
        $\psi_1 \wedge \psi_2$:
            return $check(w, \psi_1) \wedge check(w, \psi_2)$;
        $\neg\psi$:
            return $\neg check(w, \psi)$;
        $K_a\psi$:
            for $u \in R_a(w)$ do
                if $check(u, \psi) = \bot$ then
                    return $\bot$;
                endIf
            endFor
            return $\top$;
    endMatch
endFunction
```

Fig. 3. A PSPACE-algorithm for model-checking of $L^{✡1D}{}_{AGT}$

B. PSPACE-ness upper-bound of the two problems

In this subsection, we are going to give PSPACE-ness upper-bound of the model checking problem and also of the satisfiability problem. As you will see, the proof are directly given with algorithms using a polynomial amount of memory (Figures 3 and 4).

Proposition 5: Let AGT be any set of agents. The model-checking of $L^{✡1D}{}_{AGT}$ problem is in PSPACE.

Proof:

You can take a look at the recursive algorithm of Figure 3. We have to prove three points: terminaison, correctness and PSPACE-ness.

1) First let us prove terminaison by induction on φ. Let $\mathcal{T}(\varphi)$ be the property "for every world w, the call $check(w, \varphi)$ terminates".
   - $check(w, \top)$ and $check(w, p)$ terminates. So $\mathcal{T}(\top)$ and $\mathcal{T}(p)$;
   - Let us prove that $check(w, K_a\psi)$ terminates. By induction, $\mathcal{T}(\psi)$ so every call $check(u, \psi)$ terminates. So the call $check(w, K_a\psi)$ terminates and $\mathcal{T}(K_a\psi)$;
   - Other cases are treated in the same manner.
2) Secondly, we have to prove correctness. Correctness corresponds to the property $\mathcal{C}(\varphi)$ defined by "for all world w, $w \models \varphi$ iff $check(w, \varphi) = \top$". We also prove $\mathcal{C}(\varphi)$ for all formula φ by induction.
3) Finally, we prove that $check$ only requires a polynomial amount of memory. Just be careful at the line "**for** $u \in R_a(w)$ **do** ": although $R_a(w)$ may be of size exponential we do not compute it. Here we only enumerate here elements of $R_a(w)$ one by one. This can be done using only a linear amount of memory. This part is technical but I will nevertheless give some details how to implement a enumeration of elements of $R_a(w)$.
   The block:

```
for $u \in R_a(w)$ do
    if $check(u, \psi) = \bot$ then
        return $\bot$;
    endIf
endFor
```

can be rewritten in a unreadable block using a linear amount of memory in (*):

```
$u := first\_permutation(w)$
while $\neg is\_last\_permutation(u)$ do
    if $u \in R_a(w)$
        if $check(u, \psi) = \bot$ then
            return $\bot$;
        endIf
    endIf
    $u := next\_permutation(u)$;
endWhile
```

where:

- assuming we have an order $<$ over permutations of elements appearing in w, $first\_permutation(w)$ gives, using a linear amount of memory, the first permutation we can make with elements of w; For instance, if $w =$ $\underset{0}{\triangleright}\ \underset{p}{\text{☼}}$, $first\_permutation(w)$ can be $\underset{0}{\triangleright}\ \underset{p}{\blacksquare}$;
- $next\_permutation(u)$ is a function, using a linear amount of memory, giving the $<$-successor of u; For instance, we may have:
  - $next\_permutation($ $\underset{0}{\triangleright}\ \underset{p}{\blacksquare}$ $) =$ $\underset{0}{\triangleright}\ \underset{p}{\text{☼}}$;
  - $next\_permutation($ $\underset{0}{\triangleright}\ \underset{p}{\text{☼}}$ $) =$ $\underset{0}{\triangleleft}\ \underset{p}{\text{☼}}$;
  - $next\_permutation($ $\underset{0}{\triangleleft}\ \underset{p}{\text{☼}}$ $) =$ $\underset{p}{\blacksquare}\ \underset{0}{\triangleright}$ etc.
- $is\_last\_permutation(u) = \top$ iff u has no $<$-successor.

Now, we can prove by induction on φ the following property for all φ, $\mathcal{P}(\varphi)$ defined as "for all world w, the call $check(w, \varphi)$ needs $O(|\varphi| \times |w|)$ memory cells".

- $\mathcal{P}(\top)$ and $\mathcal{P}(p)$ are true;
- Let us prove $\mathcal{P}(\psi_1 \wedge \psi_2)$. The first call $check(w, \varphi_1)$ needs $O(|\varphi_1| \times |w|)$ by hypothesis of induction. Then we can release all the memory cells used for the sub-call $check(w, \varphi_1)$ and we can treat the call $check(w, \varphi_2)$. It needs $O(|\varphi_2| \times |w|)$. Hence, the sub-call $check(w, \varphi_1 \wedge \varphi_2)$ needs $max(O(|\varphi_1| \times |w|), O(|\varphi_2| \times |w|)) = O(|\varphi| \times |w|)$. So $\mathcal{P}(\psi_1 \wedge \psi_2)$.
- Now, we prove $\mathcal{P}(K_a\psi)$. By induction, every sub-call $check(w, \psi)$ needs at most $O(|\psi| \times |w|)$ memory cells. Furthermore, we need $O(|w|)$ for $first\_permutation(w)$, $is\_last\_permutation(u)$ and $next\_permutation(u)$ and also to keep the local variable u in memory. So we need, $O(|\psi| \times |w|) + O(|w|) = O(|\varphi| \times |w|)$.

Finally, $\mathcal{P}(\varphi)$ is true for all φ. In other words, the algorithm of Figure 3 only use a polynomial number of memory cells (we take in account (*)).

■

```
function $sat(\varphi)$
    $w$ := $choose\_world\_with\_symbols\_in(\varphi)$
    return $check1(w, \varphi)$
endFunction
```

Fig. 4. A PSPACE-algorithm for satisfiability problem of $L^{☼1D}{}_{AGT}$

card(AGT)	$L^{☼1D}{}_{AGT} - md$	$L^{☼1D}{}_{AGT} - sat$
1	Σ_1-hard, in Δ_2-hard	Σ_2-complete
$n \in \mathbb{N}$, $n \geq 2$	Σ_n-hard, in ??	Σ_{n+1}-hard, in ??
∞	PSPACE-complete	PSPACE-complete

card(AGT)	$S5_{card(AGT)} - sat$
1	NP-complete
$n \geq 2$	PSPACE-complete
∞	PSPACE-complete

Fig. 5. Table of complexities

Proposition 6: Let *AGT* be any set of agents. The $L^{☼1D}{}_{AGT}$-satisfiability problem is in PSPACE.

Proof: You can read the algorithm of Figure 4. The algorithm consists in guessing non-deterministically a world w and then call the routine *check* of Figure 3 to check if φ is true in w. So, the problem is NPSPACE, hence from Savitch's theorem [12], it is PSPACE. ■

Now we are going to investigate more in details complexities of the model checking and satisfiability problem depending on the size of *AGT*. The table of Figure 5 sums up all results we have. There is also the recall of complexity results about $S5_n$ satisfiability problem as comparison.

C. When AGT is infinite: PSPACE-complete

We recall the complexity result about QBF formulas satisfiability problem:

Theorem 1: The QSAT-problem defined as following:

- Input: a formula $\varphi = \exists\vec{p}_1\forall\vec{p}_2\exists\vec{p}_3\forall\vec{p}_4 \ldots Q_n\vec{p}_n\psi$ where:
  - n is any integer;
  - ψ is a boolean formula;
  - and $Q_i = \forall$ if i is even and $Q_i = \exists$ if i is odd;
  - $\vec{p}_j$ is a finite set of variables for each j.
- Output: Yes iff $\models_{QBF} \varphi$. No, otherwise.

is PSPACE-complete.

Now the following Proposition gives a translation of a QBF-instance into a $L^{☼1D}$-model-checking instance or a $L^{☼1D}$-satisfiability problem instance.

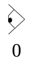
Proposition 7: Let $\varphi = \exists\vec{p}_1\forall\vec{p}_2\exists\vec{p}_3\forall\vec{p}_4\dots Q_n\vec{p}_n\psi$ be a formula of the logic QBF. We define $f(\varphi)$ by induction:

- $f(\psi) = \psi$;
- $f(\forall\vec{p}_i ... Q_n\vec{p}_n\psi) = K_{i-1}(put_i \rightarrow f(\exists\vec{p}_{i+1} ... Q_n\vec{p}_n\psi)$;
- $f(\exists\vec{p}_i ... Q_n\vec{p}_n\psi) = \hat{K}_{i-1}(put_i \wedge f(\forall\vec{p}_{i+1} ... Q_n\vec{p}_n\psi)$;

where:

- $put_a = \bigwedge_{i\in\{a+1...2n\}} \neg K^{\text{if}}_a\vec{p}_i \wedge \bigwedge_{i\in\{1...a\}} K^{\text{if}}_a\vec{p}_i$;
- $K^{\text{if}}_a\vec{p} = \bigwedge_{q\in\vec{p}} K^{\text{if}}_a q$;
- $K^{\text{if}}_a q = K_a q \vee K_a\neg q$.

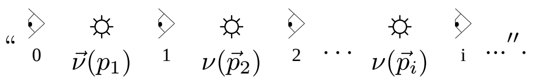
We have equivalence between:

- $\models_{QBF} \varphi$;
- $put_1 \wedge f(\forall\vec{p}_2\exists\vec{p}_3\forall\vec{p}_4\dots Q_n\vec{p}_n\psi)$ is $L^{☼1D}{}_{AGT}$-satisfiable;
- and $w \models_{L^{☼1D}} f(\varphi)$ where $w \in W_0$ where W_0 is the set of all worlds where agent 0 is completely on the left looking to the left. (we note $W_0 =$ “  ...”).

Proof: We are going to note for all $U \subseteq W$, $U \models \varphi$ iff for all $u \in U$, $u \models \varphi$. We are going to prove by induction $\models_{QBF} \varphi$ iff $W_0 \models_{L^{☼1D}} f(\varphi)$. We are going to note for all $i \in \mathbb{N}$, for all valuation $\nu[\vec{p}_1, \dots \vec{p}_i]$,

$$W_i(\nu[\vec{p}_1, \dots \vec{p}_i])$$

$$\|^{def}$$



The induction hypothesis is:

$$\nu[\vec{p}_1, \dots \vec{p}_{i-1}] \models_{QBF} Q_i\vec{p}_i \dots Q_n\vec{p}_n\psi$$

iff

$$W_{i-1}(\vec{p}_1, \dots \vec{p}_{i-1}) \models_{L^{☼1D}} f(Q_i\vec{p}_i \dots Q_n\vec{p}_n\psi)$$

.

The basis case correspond to $i = n+1$. It is the propositional case. We have:

$$\nu[\vec{p}_1, \dots \vec{p}_n] \models_{QBF} \psi \text{ iff } W_n(\vec{p}_1, \dots \vec{p}_n) \models_{L^{☼1D}} \psi.$$

Now we can attack the induction case. Let us prove for i odd. $\nu[\vec{p}_1, \dots \vec{p}_{i-1}] \models_{QBF} Q_i\vec{p}_i \dots Q_n\vec{p}_n\psi$ means that there exists a valuation $\nu(p_i)$ such that $\nu[\vec{p}_1, \dots \vec{p}_i] \models_{QBF} Q_{i+1}\vec{p}_{i+1} \dots Q_n\vec{p}_n\psi$. By induction, it means that $W_i(\vec{p}_1, \dots \vec{p}_i) \models_{L^{☼1D}} f(Q_i\vec{p}_i \dots Q_n\vec{p}_n\psi)$.

But for all $w_{i-1} \in W_{i-1}(\vec{p}_1, \dots \vec{p}_{i-1})$ and for all $w_i \in W_i(\vec{p}_1, \dots \vec{p}_i)$, we have:

- $w_{i-1}R_{i-1}w_i$;
- $w_i \models put_i$. Indeed, for all $j > i$, we have $w \models \neg K^{\text{if}}_i\vec{p}_j$ because agent j does not see lamps $\vec{p}_j$ in w_i. On the contrary, for all $j < i$, we have $w \models K^{\text{if}}_i\vec{p}_j$ because agent i do see lamps $\vec{p}_j$ in w_i (the valuation of lamps $\vec{p}_j$ is the same in all worlds $u \in R_i(w_i)$). The technical proof of $w_i \models put_i$ is left to the reader.

As $f(\exists\vec{p}_i \dots Q_n\vec{p}_n\psi) = \hat{K}_{i-1}(put_i \wedge f(\forall\vec{p}_{i+1} \dots Q_n\vec{p}_n\psi)$, we have:

$W_{i-1}(\vec{p}_1, \dots \vec{p}_{i-1}) \models_{L^{☼1D}} f(\exists\vec{p}_i \dots Q_n\vec{p}_n\psi)$. We ensure that it is equivalent.

The case where i is even is similar. ■

Immediately from this translation, we deduce the lower bound for model-checking in $L^{☼1D}$.

Corollary 2: Let *AGT* an infinite enumerable set of agents. The model-checking problem of $L^{☼1D}{}_{AGT}$ is PSPACE-hard.

Proof: Reduction via Proposition 7 and Theorem 1 in order to the PSPACE-hardness and Proposition 6. ■

In the same way we have:

Corollary 3: Let *AGT* an infinite enumerable set of agents. The satisfiability problem of $L^{☼1D}{}_{AGT}$ is PSPACE-hard.

D. When AGT is finite

We recall the complexity result about QBF formulas satisfiability problem but when the nesting of $\forall$ and $\exists$ is bounded by a fixed integer n.

Theorem 2: Let n be a integer. The QSAT$_n$-problem defined as following:

- Input: a formula $\varphi = \exists\vec{p}_1\forall\vec{p}_2\exists\vec{p}_3\forall\vec{p}_4\dots Q_n\vec{p}_n\psi$ where ψ is a boolean formula, and $Q_i = \forall$ if i is even and $Q_i = \exists$ if i is odd;
- Output: Yes iff $\models_{QBF} \varphi$. No, otherwise.

is Σ_n-complete.

The Theorem 2 only differ from Theorem 1 by the fact that n is no more a input of the problem but is now fixed inside the problem. For each integer n, we have defined the QSAT$_n$-problem. There is a enumerable number of problems.

In the same way, this precise complexity result of QBF combined with the translation of QBF to $L^{☼1D}$allows us to have complexity lower bounds of model-checking and satisfiability problem when the cardinality of the set *AGT* is finite and fixed.

Corollary 4: Let *AGT* a finite set of agents. The model-checking problem of $L^{☼1D}{}_{AGT}$ is $\Sigma_{card(AGT)}$P-hard.

Proof: Reduction via Proposition 7 and Theorem 2. ■

Corollary 5: Let *AGT* a finite set of agents. The satisfiability problem of $L^{☼1D}{}_{AGT}$ is $\Sigma_{card(AGT)+1}$P-hard.

Proof: Reduction via Proposition 7 and Theorem 2. ■

E. When card(*AGT*) = 1

Unfortunately we do not have a precise complexity upper-bound for those problems in the general case when *card*(*AGT*) is finite. Nevertheless, we have the exact complexity when *card*(*AGT*) = 1.

Proposition 8: The model-checking problem of $L^{☼1D}{}_{\{1\}}$ is in Δ_2P.

Proof: The figure 6 gives us an Δ_2P-algorithm (a P-algorithm with NP-oracles) for the model-checking problem of $L^{☼1D}{}_{\{1\}}$. Given a world w, first we compute the V of propositions occurring in φ that the agent 1 sees and I the set of propositions the agent 1 does not see. Then we can replace each occurrence p of a proposition p from V

```
function $check1(w, \varphi)$
    $V$ := set of variables that agent 1 sees in $w$;
    $I$ := set of variables that agent 1 does not see in $w$;
    $\psi := \psi$ in which we replace each $p \in V$ by $\pi_w(p)$;
    $\psi := \psi$ in which we replace each $p \in I$ not in the scope of a $K_1$ by $\pi_w(p)$;
    while there exists $K_1\chi$ subformula of $\psi$, where $\chi$ is a boolean formula do
        if $oracle - sat(\neg\chi)$ then
            $\psi := \psi$ in which we replace $K_1\chi$ by $\bot$;
        else
            $\psi := \psi$ in which we replace $K_1\chi$ by $\top$;
        endIf
    endWhile
    return $PCL(\{\bot, \top\}) - sat(\psi)$;
endFunction
```

Fig. 6. A $\Delta_2 P$-algorithm for model checking of $L^{\Sun_{1D}}{}_{\{1\}}$

```
function $sat(\varphi)$
    $w$ := $choose\_world\_with\_symbols\_in(\varphi)$
    return $check1(w, \varphi)$
endFunction
```

Fig. 7. Optimal Σ_2P-algorithm for satisfiability problem of $L^{\Sun_{1D}}{}_{\{1\}}$

in φ by the corresponding valuation $\pi_w(p)$. Concerning a proposition $p \in I$, we only replace occurrences which are not in the scope of a K_1. For instance, if $p \in I$, $q \in V$, and $\pi_w(p) = \top, \pi_w(q) = \bot$ $p \wedge q \vee K_1(p \vee q)$ is replaced by $\top \wedge \bot \vee K_1(p \vee \bot)$. Then we test satisfiability of boolean formulas $\neg\chi$ such that $K_1\chi$ is a subformula of ψ and replace $K_1\chi$ by $\bot$ if $\neg\chi$ is satisfiable and by $\top$ otherwise. At the end, we obtain a boolean formula ψ containing no variables. We test its satisfiability with $PCL(\{\bot, \top\}) - sat(\psi)$. Notice the **while** -loop is done in linear time because there are a linear number of subformulas in ψ. ■

Proposition 9: The satisfiability problem of $L^{\Sun_{1D}}{}_{\{1\}}$ is Σ_2P-complete.

Proof:

The hardness comes from Corollary 5. The Figure 7 gives us an Σ_2P-algorithm (a NP-algorithm with NP-oracles) for the satisfiability problem of $L^{\Sun_{1D}}{}_{\{1\}}$.

■

F. When AGT and ATM are both finite

Proposition 10: Let ATM a finite set of agents and $AGT = \{1\}$. The satisfiability problem and model-checking of $L^{\Sun_{1D}}{}_{\{1\}}$ is in P.

Proof: We adapt algorithms of the figure 6 and 7 in order to have an optimal polynomial algorithm. More precisely:

- You replace *choose_world_with_symbols_in* in Figure 7 by a loop over all worlds. You can notice that the set of all possible worlds is fixed, that is to say it does not depend on φ;
- $oracle - sat$ can now run in polynomial time because there is a fixed number of propositions.

■

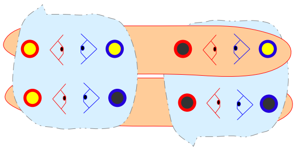


Fig. 8. World for Muddy children

IV. Public announcements

As done in [11] we can extend our framework with public announcements. This is essentially motivated by modelling examples like Muddy children. With public announcements, an agent will be able to learn something about the part of the actual world which he can not see. The technique is classical: we add an operator $[\varphi!]$ and we define semantics as in $S5_n$.

A. Definitions

Our new language $\mathcal{L}^!_{AGT}$ is defined by the following BNF:

$$\varphi ::= p \mid \varphi \wedge \varphi \mid \neg\varphi \mid K_a\psi \mid [\varphi!]\varphi$$

where $p \in ATM$ and $a \in AGT$.

From now, we do not only parameter $\models$ with a world but also with the set of worlds.

Definition 12 (truth conditions): Let U a set of worlds ($U \subseteq W$). Let $w \in U$. We define $U, w \models \varphi$ by induction:

- $U, w \models p$ iff $\pi(p) = \top$
- $U, w \models \varphi \wedge \psi$ iff $U, w \models \varphi$ and $U, w \models \psi$;
- $U, w \models \neg\varphi$ iff $U, w \not\models \varphi$;
- $U, w \models K_a\psi$ iff for all $w' \in U$, wR_aw' implies $U, w' \models \psi$;
- $U, w \models [\varphi!]\psi$ iff $U, w \models \varphi$ implies $U \cap \{w' \in U \mid U, w' \models \varphi\}, w \models \psi$.

The set of validities we obtain is noted $L^{\Sun_{1D}!} = \{\varphi \mid W, w \models \varphi\}$ where W is the set of all worlds defined Definition 8.

B. Example

Now we are going to study the Muddy children example. This example is also studied in [11]. You can also find this example in [4] and [3] with more than two children. The situation is the following: there are two children named 1 and 2. Their foreheads are dirty. They see each others. The situation is represented by the world w shown in Figure 8 in the top left. One child do not know if he is dirty or not but he knows the state of the forehead of the other one. We introduce two propositions: p stands for “1’s forehead is dirty” and q stands for “1’s forehead is dirty”.

We have:

- $W, w \models K_1q \wedge K_2p$;
- $W, w \models \neg K_1p \wedge \neg K_1(\neg p) \wedge \neg K_2(\neg q)$.

Then:

- the father says at least one of them are dirty;

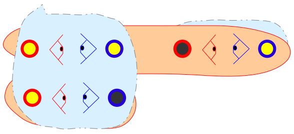


Fig. 9. World for Muddy children after having announced φ_1

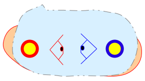


Fig. 10. World for Muddy children after having announced φ_1 and φ_2

- the children answer that they do not know whether they are dirty of not.

Formally, we also have:

$$W, w \models [\varphi_1!][\varphi_2!](K_1 p \wedge K_2 q)$$

where:

- $\varphi_1 = p \vee q$;
- $\varphi_2 = (\neg K_1 p \vee \neg K_1 \neg p) \wedge (\neg K_2 q \vee \neg K_2 \neg q)$.

We verify that after having announced φ_1, we only consider worlds in Figure 9. Then we only consider the initial world w drawn in 10.

C. Complexity

Because $L^{\sun_{1D}!}$ is a conservative extension of $L^{\sun_{1D}}$, we inherit from the lower bound results both for model-checking and satisfiability. In fact, we keep the PSPACE-ness upper-bound with public announcements.

Proposition 11: The model-checking and satisfiability problem in $L^{\sun_{1D}!}$ is PSPACE-complete.

Proof: The Figure 11 gives an algorithm for model-checking. As usual, w is a world, φ is a formula. The second argument C is a list of formulas and stands for the *context*: if $C = []$ (empty list), it corresponds to the whole set of worlds otherwise it is a list of announced formulas used to update the model. More precisely, let us define:

- $W^{[]} = W$;
- $W^{[\psi_1 : C']} = \{w \in W^{C'} \mid W^{C'}, w \models \psi_1\}$.

We want:

- $mc(w, C, \varphi)$ returns true iff $W^C, w \models \varphi$;
- $inupdatedM(w, C)$ returns true iff $w \in W^C$.

We have to prove terminaison, correctness and complexity. Let us begin to prove terminaison. First of all, we are going to introduce an order $\prec$ over all possible inputs (C, φ) of the function $mc!$ of Figure 11.

We define $(C, \varphi) \prec (D, \psi)$ by:

- $|C| + |\varphi| < |D| + |\psi|$;
- or $\left\{ \begin{array}{l} |C| + |\varphi| = |D| + |\psi| \\ \text{and } |\varphi| < |\psi| \end{array} \right.$

where

```
function $mc!(w, C, \varphi)$
    match $(\varphi)$
        $\top$:
            return $\top$;
        $p \in ATM$:
            return $\top$ if $p$ is true in $w$;
            return $\bot$ if $p$ is false in $w$;
        $\psi_1 \wedge \psi_2$:
            return $mc!(w, C, \psi_1) \wedge mc!(w, C, \psi_2)$;
        $\neg\psi$:
            return $\neg mc!(w, C, \psi)$;
        $K_a \psi$:
            for $u \in R_a(w)$ do
                if $inupdatedM(u, C)$ and $mc!(u, \psi) = \bot$ then
                    return $\bot$;
                endIf
            endFor
            return $\top$;
        $[\psi_1!]\psi_2$:
            if $mc!(w, C, \psi_1)$ then
                return $mc!(w, [\psi_1 : C], \psi_2)$
            else
                return $\top$
            endIf
    endMatch
endFunction
```

Fig. 11. Algorithm for model-checking in $L^{\sun_{1D}!}$

- $|\varphi|$, $|\psi|$ denotes the length (number of symbols) in φ, ψ;
- $|C|$, $|D|$ denotes also the number of symbols in C, D. More precisely:
  - $|[]| = 0$;
  - $|[\psi_1 : C']| = |\psi_1| + |C'|$.

The order $\prec$ is well-founded and we can use it to prove terminaison by induction.

- Basic case: $(\emptyset, p)$ etc.
- Induction case: you just have to see that $mc!(w, C, \varphi)$ will only call $mc!(u, D, \psi)$ with $(D, \psi) \prec (C, \varphi)$. For instance, when $mc!(w, C, \varphi)$ calls $inupdatedM(u, C)$ which calls $mc!(w, C', \psi)$, we have $|C'| + |\psi| = |C| < |C| + |\varphi|$.

Correction and the fact that the algorithm runs using a polynomial space can also be proved by induction using the order $\prec$.

The hardness comes from the fact that $L^{\sun_{1D}!}$ is a conservative extension of $L^{\sun_{1D}}$ and the model checking of $L^{\sun_{1D}}$ is PSPACE-hard (Corollary 2).

Concerning the satisfiability, we can make the same remark than in the proof of Proposition 6.

■

The upper-bound in special cases (*AGT* finite etc.) has not been studied yet.

From now, we are to discuss about the implementation and develop the example of the Muddy children.

```
function $inpudatedM(w, C)$
    match $(C)$
        $[]$: return $\top$;
        $[\psi : C']$: $mc!(w, C', \psi)$;
    endMatch
endFunction
```

Fig. 12. Algorithm for testing if a world w is in the updated model formulas in C

V. Implementation

You can find an implementation on the Web site. You can put positions and directions of agents and positions and states of lamps on your own. Then you can write down a formula and check if your formula is true in the world you have drawn.

This program offers a concrete example to illustrate epistemic logic to students.

A. Description

The program is written in Scheme for the easy use of data structures and recursive programming. Haskell could also be a well-suited language especially for the lazy evaluation enabling us to write a program which seems to use a exponential amount of memory whereas it uses only a polynomial amount of memory. Here are the main Scheme functions:

- `(mc world formula)` computes if the formula `formula` is true in the world `world`;
- `(mc-with-context world context formula)` computes of the formula `formula` is true in the world `world` but we restrict our check computations only on worlds satisfying the formula `context`;
- `(worldset-delete-not-satisfying worldset formula)` removes from the set of world `worldset` all worlds which does not satisfy the formula `formula`. This function is used to deal with updated models;
- `(world-getpossibleworlds world agent)` computes the set of all possible worlds for agent `agent` in world `world`.

In order to be human readable, the implementation does not run in polynomial space but in exponential time. For instance the function `(world-getpossibleworlds w a)` computes really *all* worlds in $R_a(w)$.

B. Practising Muddy children

You can describe the *current situation* (world w on the top left in Figure 8 by `( (p #t) (1 <) (2 >) (q #t))`. Notice that we are not going to construct the Kripke structure by hand. When you draw a Kripke model, you can easily mistakes all the more so the model is theoretical. Here we just enjoy specifying *graphically* the situation. The Kripke structure is then generated on-the-fly by the algorithm. You can test if $W, w \models K_1 p \wedge K_2 q$ by calling

```
(mc '( (p #t) (1 <) (2 >) (q #t)) ((1
knows p) and (2 knows q))).
```

The function returns `#f` meaning that we do not have $W, w \models K_1 p \wedge K_2 q$.

We ask the computer the different worlds the agent 1 imagine. To do this we write

```
(world-getpossibleworlds '( (p #t) (1 <)
(2 >) (q #t)) 1).
```

The system gives:

```
(((p #t) (1 <) (2 >) (q #t))
((p #f) (1 <) (2 >) (q #t))).
```

We can now test if the formula $W, w \models [\varphi_1!][\varphi_2!](K_1 p \wedge K_2 q)$. You simply write

```
(mc '( (p #t) (1 <) (2 >) (q #t))
(announce (p or q) (announce ((not (1
knows p)) and (not (2 knows q))) ((1 knows
p) and (2 knows q)))))
```

The system answers `#t`.

VI. Conclusion

The epistemic logic $S5_n$ is a general and theoretical framework for the representation of knowledge. In this paper, we have studied a spatially grounded epistemic logic. We have investigated two aspects of knowledge learning:

- With $L^{\sun 1D}$, we can reason about what agents know by learning only with their eyes (when they are located on a line space);
- With $L^{\sun 1D^{!}}$ we can reason about what agents know by looking at their environments and by listening to public announcements.

Of course the case of the line is restrictive. The case of the plane or of the space may be more interesting. Nevertheless, this paper gives complexity results for model-checking and satisfiability problem for the case of the line. Even the line looks like easy, problems are already PSPACE-complete if the number of agents is not bounded. We conjecture that the complexity of this logic for dimension $n \geq 2$ remains PSPACE-complete.

From now, there are two main perspectives: to adapt this logic to the case of two dimensions [1] and to study properly complexity of model checking and satisfiability with/without public announcements. Other perspectives are numerous:

- fill the Figure 5. The exact complexity classes of model checking and satisfiability ${L^{\sun 1D}}_{AGT}$ when *AGT* is finite are still open questions;
- Study and implement the logic with agents and lamps in the plane [1] and compare it to the logic in the line. Writing down the semantics is quite easy: you just have to replace $\mathbb{R}$ by $\mathbb{R}^2$ in Definition 8 and tune the definition of directions and Definition 3. The main difficulty is to find a compact representation in order to deal with the model checking and satisfiability problem. In two dimensions it is no more possible to consider a total preorder on elements. Finding a good equivalent of Definition 7 satisfying Proposition 4 in the case of dimension 2 or more is still an open problem.
- Study the logic in the 3D-space and compare it to the one in the plane (I guess we obtain the same validities);

- Find an axiomatization of those logics in order to understand better how they work;
- Study if it is possible to have a normal form (like for S5, where all formulas are equivalent to a formula of modal degree 1 [6]);
- Extend with a common knowledge operator. Will the complexity of the satisfiability problem also increase and become EXPTIME-complete?
- Extend with private communications between agents.

Acknowledgements.
I thank Philippe Balbiani, Olivier Gasquet, Andreas Herzig, Emiliano Lorini and the reviewers for their different helpful remarks.

References

[1] Edwin Abbott Abbott. *Flatland*. Basil Blackwell, 1884.

[2] Jon Barwise and John Etchemendy. *Tarski's World: Version 4.0 for Macintosh (Center for the Study of Language and Information - Lecture Notes)*. Center for the Study of Language and Information/SRI, 1993.

[3] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.

[4] V. Goranko and M. Otto. *Handbook of Modal Logic*, chapter Model Theory of Modal Logic, pages 255–325. Elsevier, 2006.

[5] Joseph Y. Halpern and Yoram Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(3):319–379, 1992.

[6] G. E. Hughes and M. J. Cresswell. *An Introduction to Modal Logic*. Methuen and Co., 1968.

[7] Ethan Kennerly, Andreas Witzel, and Jonathan A. Zvesper. Thief belief (extended abstract). Presented at Logic and the Simulation of Interaction and Reasoning 2 (LSIR2) Workshop at IJCAI-09, July 2009.

[8] E. Lorini, L. Tummolini, and A. Herzig. Establishing mutual beliefs by joint attention: towards a formal model of public events. In Monica Bucciarelli Bruno G. Bara, Lawrence Barsalou, editor, *Proceedings of the XXVII Annual Conference of the Cognitive Science Society*, pages 1325–1330. Lawrence Erlbaum, 2005.

[9] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[10] Rohit Parikh, Lawrence S. Moss, and Chris Steinsvold. Topology and epistemic logic. In *Handbook of Spatial Logics*, pages 299–341. 2007.

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

[12] Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *J. Comput. Syst. Sci.*, 4(2):177–192, 1970.

[13] Hans P. van Ditmarsch, Wiebe van der Hoek, Ron van der Meyden, and Ji Ruan. Model checking russian cards. *Electr. Notes Theor. Comput. Sci.*, 149(2):105–123, 2006.