How to share knowledge by gossiping

Andreas Herzig, Faustine Maffre

How to share knowledge by gossiping

Andreas Herzig and Faustine Maffre
University of Toulouse, IRIT, 118, Route de Narbonne, F-31062 Toulouse, France

We provide a logical investigation of a simple case of communication in a network of agents called the gossip problem. Its classical version is: given n agents each of which has a secret—a fact not known to anybody else—, how many calls does it take to achieve shared knowledge of all secrets, i.e., to reach a state where every agent knows every secret? Several protocols achieving shared knowledge in $2(n-2)$ calls exist and were proved to be optimal: no shorter sequence of calls exists. We generalize that problem and focus on higher-order shared knowledge: how many calls does it take to obtain that *everybody knows* that everybody knows all secrets? More generally, how many calls does it take to obtain shared knowledge of order k? This cannot be achieved simply by communicating facts: the agents also have to communicate higher-order knowledge of facts. We give an algorithm that works in $(k+1)(n-2)$ calls. We analyse its properties in a logic that we have investigated in previous work and that is based on the concept of observability of propositional variables by agents: Dynamic Epistemic Logic of Propositional Assignment and Observation DEL-PAO. This enables us in particular to give a formal proof of correctness of the algorithm.

Keywords: gossip protocol, epistemic logic, shared knowledge, common knowledge, theory of mind, dynamic epistemic logic, visibility, observability

1. Introduction: the gossip problem and its generalization

The gossip problem is typically introduced in the literature in the following terms [1,28]:

> There are six agents each of which knows some secret not known to anybody else. Two agents can make a telephone call and exchange all secrets they know. How many calls does it take to share all secrets, i.e., how many calls have to take place until everybody knows all secrets?

The problem can be generalized to an arbitrary numbers of agents $n \geq 2$. A straightforward solution has one call per couple of agents (i, j). This takes $n(n-1)$ calls. But one can do much better.

To warm up consider $n=3$. Let s_i denote agent i's secret. Shared knowledge can be attained in three calls, as can be seen by inspecting the following protocol:

1. 1 calls 2: 1 tells s_1, and 2 tells s_2;
2. 1 calls 3: 1 tells s_1 *and* s_2, and 3 tells s_3;
3. 1 calls 2: 1 tells s_3.

After the second call, both 1 and 3 know $s_1 \wedge s_2 \wedge s_3$. So 1 and 3 are both *experts*. However, 2 is not an expert yet because she does not know 3's secret: she only becomes an expert after the third call.

While the cases $n=2$ and $n=3$ are special, for $n \geq 4$ there is a general result saying that shared knowledge can be achieved in $2(n-2)$ calls, and that it is impossible to do better: there is no protocol with less than $2(n-2)$ calls [8,32,25].

In the present paper we are interested in contexts where the goal is to achieve higher-order shared knowledge: how many calls does it take to obtain that *everybody knows* that everybody knows all secrets? More generally, how many calls does it take to obtain shared knowledge of order k? Such higher-order knowledge is important for social intelligence: social interaction requires the ability to reason about the other agents' mental states. In other words, an agent should entertain a *theory of mind*: she should have beliefs about other agents' beliefs, beliefs about what other agents believe about her, etc. Epistemic logic is an interesting tool in the investigation of theory of mind; see e.g. [12,24] for recent work on the subject.

If we make the hypothesis that there is a global clock and that everybody knows the protocol then things are easy: $2(n-2)$ calls achieve not only shared knowledge of all secrets, but also common knowledge of all secrets. Here we do not make that hypothesis and investigate asynchronous contexts. In such contexts, for $n = 2$ it still holds that a single call makes all secrets not only shared knowledge, but also common knowledge; however, for $n > 2$ common knowledge cannot be achieved by a finite number of calls.

We are therefore more modest and consider the goal of shared knowledge of order k. The original problem

is to reach shared knowledge of order k=1. Now consider k=2. There, the agents cannot achieve the goal by communicating facts alone: they also have to communicate what they know about the facts. Let us illustrate this for n=3 agents. We modify the protocol for shared knowledge of order 1 and let the agents communicate knowledge instead of facts:

1. 1 calls 2: 1 tells K_1s_1, and 2 tells K_2s_2;
2. 1 calls 3: 1 tells K_1s_1, K_1s_2, K_2s_1 *and* K_2s_2, and 3 tells K_3s_3;
3. 1 calls 2: 1 tells K_1s_3, K_3s_1, K_3s_2 *and* K_3s_3,

where $K_i\varphi$ reads "agent i knows that φ is true". After the second call, 1 and 3 are both *experts of level* 1: both know $s_1 \wedge s_2 \wedge s_3$. However, none of them is an *expert of level* 2 because none of them knows K_2s_3 (and so for a good reason: K_2s_3 is not the case yet). The third call makes both 1 and 2 experts of level 2. However, agent 3 is not an expert of level 2 yet: although K_2s_3 is the case, agent 3 does not know this. To attain that level it takes a further, fourth call between 1 and 3 (or, alternatively, between 2 and 3). This shows that the first three calls are not enough to obtain shared knowledge of order 2.

Our example illustrates that in order to attain shared knowledge of level 2, it is useful to communicate knowledge about facts (that is, knowledge of level 1). It also illustrates that it takes more calls to achieve shared knowledge of level $k+1$ than it takes to achieve shared knowledge of level k.

We note $Gossip(k,n)$ the instance of the generalized gossip problem with $n \geq 2$ agents and the goal to achieve depth $k \geq 1$ of shared knowledge. So the original problem corresponds to the instance $Gossip(1,6)$. A solution to $Gossip(k,n)$ is a sequence of calls between agents resulting in shared knowledge of depth k of all secrets by all agents. We are going to introduce an algorithm solving $Gossip(k,n)$ in $(k+1)(n-2)$ calls, for $n \geq 4$. Based on results in [18], we moreover establish that our algorithm is optimal: at least $(k+1)(n-2)$ calls are necessary to achieve the goal of $Gossip(k,n)$. Our proofs are formally rigorous: they are couched in a dynamic epistemic logic that is called DEL-PAO (Dynamic Epistemic Logic of Propositional Assignment and Observation), with epistemic operators K_i, for i an agent, and dynamic operators $[Call^i_j]$, for i and j two different agents. Building on previous work on observability-based knowledge [31,36,35], we had introduced and studied that logic in [26].

The paper is organized as follows. Section 2 defines some notation. Section 3 presents our algorithm for $n \geq 4$ agents. Section 4 recalls syntax and semantics of the dynamic epistemic logic DEL-PAO. In Section 5 we show how to capture the algorithm as a DEL-PAO program. In Section 6 we prove in DEL-PAO that the algorithm is correct and in Section 7 that our algorithm is optimal. In Section 8 we study the special cases of two and three agents and in Section 9 we briefly discuss a version of the problem with goals involving ignorance. Section 10 concludes.

2. Notation

In this section we introduce some notation in order to be able to formally talk about the agents' knowledge and about gossip protocols. In Section 4 we are going introduce a full-fledged logical language into which that notation can be embedded.

Let $Agt = \{1,\ldots,n\}$ be the set of all agents. The secret of agent i is denoted by s_i. The set of propositional variables is $Prop = \{s_i : i \in Agt\}$. To simplify things we suppose that s_i is true. A more general framework where secrets can be true or false can be found in [5,6].

2.1. Notation for knowledge

We write $K_i\varphi$ to express that agent i knows that the formula φ is true.

The initial situation before the agents start gossiping is expressed by

$$\bigwedge_{i\in Agt}\Big(s_i \wedge K_i s_i \wedge \bigwedge_{j\in Agt, j\neq i}\Big(\neg K_j s_i \wedge \neg K_j \neg s_i\Big)\Big).$$

The goal of shared knowledge of level 1 is expressed by the formula

$$\bigwedge_{i\in Agt} K_i\Big(\bigwedge_{j\in Agt} s_j\Big)$$

which says that every agent knows every secret. Letting s_J abbreviate the conjunction $\bigwedge_{i\in J} s_i$ of secrets of agents in J, we can write this more compactly as

$$\bigwedge_{i\in Agt} K_i s_{Agt}.$$

2.2. Notation for shared knowledge

Let $EK_J\varphi$ denote the conjunction $\bigwedge_{i\in J} K_i\varphi$, for non-empty sets of agents $J \subseteq Agt$. It describes situations where φ is shared knowledge in J: each agent in J knows that φ.

We can now describe several situations of shared knowledge in a more convenient way:

- $EK_{Agt}s_{Agt}$ expresses that all secrets are shared knowledge;
- $EK_{Agt}EK_{Agt}s_{Agt}$ expresses that every agent knows that all secrets are shared knowledge;
- $\underbrace{EK_{Agt}\ldots EK_{Agt}}_{k \text{ times}}s_{Agt}$ expresses that all secrets are shared knowledge up to depth $k \geq 1$.

Let us also introduce an abbreviation for shared knowledge of level m, for $m \geq 0$: we inductively define $EK_J^0\varphi = \varphi$ and $EK_J^{m+1}\varphi = EK_JEK_J^m\varphi$. So the goal of shared knowledge of level k of all agents can be written

$$EK^k_{Agt}s_{Agt}.$$

We drop set parentheses and write $s_{i_1,\ldots,i_m}$ instead of $s_{\{i_1,\ldots,i_m\}}$ and $EK_{i_1,\ldots,i_m}$ instead of $EK_{\{i_1,\ldots,i_m\}}$.

2.3. Notation for calls and gossip protocols

We express calls and their consequences by means of modal operators of action as familiar from dynamic logic. The action $Call^i_j$ expresses that i and j talk to each other. (It does not matter who initiates the call.) The formula $[Call^i_j]\varphi$ expresses that φ is true after i and j talked to each other. So the formula

$$[Call^i_j]EK_{\{i,j\}}s_{\{i,j\}}$$

expresses that the result of $Call^i_j$ is that the secrets of i and j become shared knowledge of the group $\{i,j\}$. With the above convention this can be written $[Call^i_j]EK_{i,j}s_{i,j}$. More generally, a call between i and j achieves common knowledge between i and j: we have

$$[Call^i_j]EK^m_{i,j}s_{i,j}$$

for arbitrary m.

In the sequence of calls

$$Call^{i_1}_{j_1};\ldots;Call^{i_m}_{j_m},$$

i_1 calls j_1 first, then i_2 calls j_2, …, and finally i_m calls j_m. The length of the sequence is m.

An instance of the generalized gossip problem, written $Gossip(k,n)$, is given by a number of agents n and the level k of shared knowledge to be attained. A solution of $Gossip(k,n)$ is a sequence of calls π such that

$$\bigwedge_{i\in Agt}\Big(s_i \wedge K_is_i\Big) \rightarrow [\pi]EK^k_{Agt}s_{Agt}$$

holds.

The above statements should be intuitively clear, even we postpone the semantics of our language to Section 4. Readers familiar with dynamic epistemic logics may take the epistemic operators K_i to be operators of the standard epistemic logic S5 and may take the actions $Call^i_j$ as private announcements made to i and j [11,23,22,10,37]. The semantics we adopt is going to be conceptually and mathematically simpler than those of dynamic epistemic logics.

3. An algorithm achieving shared knowledge of depth k

Let $Gossip(k,n)$ be an instance of the generalized gossip problem, for some $k\geq 1$ and $n\geq 4$. The following algorithm generates a sequence of $(k+1)(n-2)$ calls. Throughout the algorithm, two of the agents, called *left* and *right*, will play a central, fixed role: each of the other agents only communicates with either *left* or *right*. The $n-2$ remaining agents will be numbered 0, 1, …, $n-3$.

The algorithm is made up of *turns*. During each turn, *left* and *right* collect the secrets of other agents. Together with the last agent they talked to in that turn, they thereby become what we call ‘semi-experts’. A further call between complementary semi-experts turns them into full experts. The last agents to which *left* and *right* talked also play a crucial role. These two further semi-experts alternate: they are permuted at each turn in a way that will guarantee that the goal is reached.

Algorithm 1. *For $t = 0..k$ do*

agent left calls agent $0-t \pmod{n-2}$*;*
agent left calls agent $1-t \pmod{n-2}$*;*
$\vdots$
agent left calls agent $n-3$*;*
agent left calls agent 0*;*
agent left calls agent 1*;*
$\vdots$
agent left calls agent $n-4-t \pmod{n-2}$*;*
agent right calls agent $n-3-t \pmod{n-2}$.

In words:

- At the first turn (turn 0), agent *left* calls agent 0, then 1, ..., then $n-4$, and finally agent *right* calls agent $n-3$;
- At the second turn (turn 1), agent *left* calls agent $n-3$, then 0, then 4, ..., then $n-5$, and finally agent *right* calls agent $n-4$;
- ... and so on.

So each turn involves $n-2$ calls, and overall the algorithm produces a sequence of $(k+1)(n-2)$ calls.

In the rest of the paper, we assume that every agent index is taken modulo $n-2$ and omit "$\pmod{n-2}$".

Figure 1 gives a visual representation of Algorithm 1: agents $0, 1, \ldots, n-3$ are put on a wheel which, between each turn, rotates clockwise. Agent *left* calls everybody in ascending order—his sequence of calls is depicted by an orange arrow—, except the agent at the rightmost position of the wheel, then *right* (sequence of calls in blue) calls this agent.

Theorem 1. *The minimal number of calls needed to solve the instance Gossip(k,n) of the generalized gossip problem, for $k \geq 1$ and $n \geq 4$, is $(k+1)(n-2)$.*

The main part of the paper is devoted to the proof of the above theorem. In sections 6 and 7 we prove that the sequence of calls produced by the algorithm is indeed a solution and that the goal cannot be achieved in less calls. Our proofs will be done in the formal language of the logic DEL-PAO; in the next two sections we introduce the logic and show how to model calls within its language.

4. Dynamic Epistemic Logic of Propositional Assignment and Observation DEL-PAO

Dynamic Epistemic Logic of Propositional Assignment and Observation DEL-PAO is grounded on the notion of observability of propositional variables. It refines a logic that was proposed and studied in a series of papers by van der Hoek, Wooldridge and colleagues under the names Epistemic Coalition Logic of Propositional Control with Partial Observability ECL-PC(PO) [36] and Logic of Revelation and Concealment LRC [35]. These logics stem from languages used in model checkers such as MOCHA in order to compactly describe a distributed system [2,31,33]. The idea is that each agent has a set of propositional variables she can observe: no different truth value is possible for her. The other way round, any combination of truth values of the non-observable variables is possible for her. We recall this logic now; more details are in [26]. Further developments of our work are reported in [14,16,27]. It can also be related to approaches which aim at grounding knowledge on a spatial notion of visibility [9,21].

4.1. Observability atoms

The atomic formulas of DEL-PAO are called *visibility atoms* and take the form $\mathsf{S}_{i_1}\mathsf{S}_{i_2}...\mathsf{S}_{i_m}p$, where p is a propositional variable from a countable non-empty set *Prop* and $i_1, i_2, ..., i_m$ are agents from a finite non-empty set *Agt*. When m=0 then we have nothing but a propositional variable. For m=1, the atom $\mathsf{S}_{i_1}p$ reads "agent i_1 sees the value of the variable p" or "agent i_1 sees whether p is true or not", and for m=2, the second-order observation $\mathsf{S}_{i_1}\mathsf{S}_{i_2}p$ reads "agent i_1 sees whether i_2 sees the value of p"; and so on. Beyond individual observability the language of DEL-PAO also accounts for joint observability: the atom $\mathsf{JS}p$ reads "all agents jointly see the value of p". Metaphorically, joint attention about p is the case when there is eye contact between the agents when observing p. Joint visibility implies individual visibility: when $\mathsf{JS}p$ is true then $\mathsf{S}_i p$ should also be true.

One can define first- and higher-order knowledge about literals by means of conjunctions of visibility atoms. Indeed, for a propositional variable p we have that agent i knows that p is true when p is true and i sees p. Similarly i knows that p is false when p is false and i sees p. The list below collects some equivalences that will be valid:

$$K_i p \leftrightarrow p \wedge \mathsf{S}_i p$$
$$K_i \neg p \leftrightarrow \neg p \wedge \mathsf{S}_i p$$
$$\neg K_i p \wedge \neg K_i \neg p \leftrightarrow \neg \mathsf{S}_i p$$
$$K_j K_i p \leftrightarrow p \wedge \mathsf{S}_i p \wedge \mathsf{S}_j p \wedge \mathsf{S}_j \mathsf{S}_i p$$
$$K_j K_i \neg p \leftrightarrow \neg p \wedge \mathsf{S}_i p \wedge \mathsf{S}_j p \wedge \mathsf{S}_j \mathsf{S}_i p$$

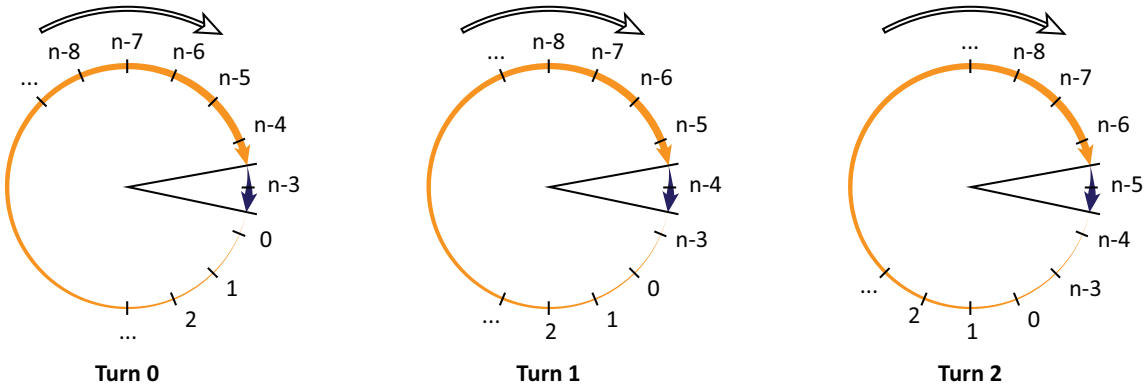

Fig. 1. Graphical representation of the first three turns of Algorithm 1.

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

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

The elements of *ATM* are also called *visibility atoms*, or atoms for short. For example, $\mathtt{JS}\,\mathtt{S}_2\, q$ reads "all agents jointly see whether agent 2 sees the value of q"; in other words, there is joint attention in the group of all agents concerning 2's observation of q. The elements of *ATM* are noted $\alpha, \alpha', \ldots, \beta, \beta', \ldots$.

4.2. Complex formulas

Beyond atomic formulas the language of DEL-PAO has epistemic operators as well as actions, alias programs, assigning truth values to visibility atoms. It is defined by the following grammar:

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

where α ranges over *ATM* and i over *Agt*.

Our atomic programs are assignments of truth values to atoms from *ATM*: $+\alpha$ makes α true and $-\alpha$ makes α false. Complex programs are constructed with dynamic logic operators: $\pi;\pi'$ is sequential composition ("first execute π, then π'"), $\pi \sqcup \pi'$ is non-deterministic choice ("execute either π or π', choosing non-deterministically"), and $\varphi?$ is test ("if φ is true then continue the execution, else stop"). Just as in dynamic logic, the formula $[\pi]\varphi$ reads "after every execution of π, φ is true". The formula $K_i\varphi$ reads "i knows that φ is true on the basis of what she observes", and $CK\varphi$ reads "all agents jointly know that φ is true on the basis of what they jointly observe". These epistemic operators account for forms of individual and common knowledge that are respectively obtained via individual observation and joint observation of facts. They therefore differ conceptually from the classical operators of individual and common knowledge as studied in the area of epistemic logic [20].

The other boolean operators $\top$, $\bot$, $\vee$, $\rightarrow$ and $\leftrightarrow$ are defined as abbreviations, and $\langle\pi\rangle\varphi$ ("there exists an execution of π after which φ is true") abbreviates $\neg[\pi]\neg\varphi$. Moreover, *skip* abbreviates $\top?$ and *fail* abbreviates $\bot?$. Finally, **if** φ **then** π abbreviates $(\varphi?;\pi) \sqcup \neg\varphi?$.

The set of atomic formulas of *ATM* occurring in the formula φ is noted $ATM(\varphi)$; the set $ATM(\pi)$ is defined similarly. For example, if $\pi = q?;+\mathtt{S}_2\, p$ and $\varphi = [\pi]\mathtt{S}_1\,\mathtt{JS}\, p \rightarrow q$, then $ATM(\pi) = \{q, \mathtt{S}_2\, p\}$ and $ATM(\varphi) = \{q, \mathtt{S}_2\, p, \mathtt{S}_1\,\mathtt{JS}\, p\}$. (So $\mathtt{JS}\, p \notin ATM(\varphi)$.)

4.3. Introspective valuations

The models of DEL-PAO are simply sets of visibility atoms. In order to guarantee positive and negative introspection we have to ensure that agents are always aware of what they see: for every agent i and propositional variable p, $\mathtt{S}_i\,\mathtt{S}_i\, p$ has to be in every valuation. More generally, a valuation V is introspective when it contains every visibility atom having two consecutive $\mathtt{S}_i$, such as $\mathtt{S}_j\,\mathtt{S}_i\,\mathtt{S}_i\,\mathtt{S}_k\, p$. So in an introspective valuation an agent is aware of what she sees, every agent sees this, and every agent sees that every agent sees this, etc.

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

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

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

(C1) is about introspection of individual sight: an agent always sees whether she sees the value of an atom. (C2) requires the same for joint sight; indeed, if $\mathtt{JS}\,\alpha$ is true then $\mathtt{JS}\,\mathtt{JS}\,\alpha$ should be true by introspection, and if $\mathtt{JS}\,\alpha$ is false then all agents jointly see that at least one of them has broken eye contact. (C3) forces the first to be common knowledge. (C4) guarantees that joint visibility implies individual visibility. Constraints (C4) and (C5) guarantee that $\mathtt{JS}\,\alpha \in V$ implies $\sigma\,\alpha \in V$ for $\sigma \in OBS^+$. This motivates the following relation of *introspective consequence* between atoms: $\alpha \rightsquigarrow \beta$ iff either $\alpha = \beta$, or $\alpha = \mathtt{JS}\,\alpha'$ and $\beta = \sigma\,\alpha'$ for some σ.

Proposition 1 ([26])**.** *A valuation $V \in 2^{ATM}$ is introspective if and only if, for every $\alpha, \beta \in ATM$ and $i \in Agt$:*

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

An atom $\alpha \in ATM$ is *valid in INTR* if and only if α belongs to every valuation in *INTR*. By Proposition 1, α is valid in *INTR* if and only if α is of the form either $\sigma\,\mathtt{S}_i\,\mathtt{S}_i\,\alpha$ with $\sigma \in OBS^*$, or $\sigma\,\mathtt{JS}\,\alpha$ with $\sigma \in OBS^+$.

4.3.1. Indistinguishability relations between valuations

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

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

where we write $V(\alpha) = V'(\alpha)$ when α has the same truth value in V and V', i.e., when either $\alpha \in V$ and $\alpha \in V'$, or $\alpha \notin V$ and $\alpha \notin V'$.

It is proven in [26] that $\sim_i$ and $\sim_{Agt}$ are equivalence relations on the set of introspective valuations *INTR* and that no valuation of *INTR* is $\sim_i$- or $\sim_{Agt}$-related to valuations outside of *INTR*.

4.3.2. Truth conditions and validity

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

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

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

Now the truth conditions are as follows:

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

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

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

The relation R_π is defined just as in PDL for the program operators ;, $\sqcup$ and ?. The interpretation of assignments is designed in a way such that we stay in *INTR*: the program $+\alpha$ adds all the positive consequences of α; the program $-\alpha$ fails if α is valid in *INTR* and otherwise removes all the negative consequences of α. For example, we never have $V R_{-\mathtt{S}_1\,\mathtt{S}_1\,p} V'$, i.e., the program $-\mathtt{S}_1\,\mathtt{S}_1\,p$ always fails.

In contrast, the program $-\mathtt{S}_1\,\mathtt{S}_2\,p$ always succeeds, and we have $VR_{-\mathtt{S}_1\,\mathtt{S}_2\,p}$ $(V \setminus \{\mathtt{S}_1\,\mathtt{S}_2\,p, \mathtt{JS}\,\mathtt{S}_2\,p, \mathtt{JS}\,p\})$ because the only atoms—beyond $\mathtt{S}_1\,\mathtt{S}_2\,p$ itself—whose consequence is $\mathtt{S}_1\,\mathtt{S}_2\,p$ are $\mathtt{JS}\,\mathtt{S}_2\,p$ and $\mathtt{JS}\,p$. Therefore $V \not\models [-\mathtt{S}_1\,\mathtt{S}_2\,p]\mathtt{JS}\,p$ for every V.

Similarly to the relations $\sim_i$ and $\sim_{Agt}$, it is proven in [26] that each R_π only relates valuations in *INTR* to valuations in *INTR*. Therefore there is no risk to leave the set of introspective valuations when interpreting a modal operator.

When $V \models \varphi$ we say that V is a *model* of φ. The set of (not necessarily introspective) models of φ is noted $\|\varphi\|$. A formula φ is *satisfiable in INTR* if φ has an introspective model, i.e., if $\|\varphi\| \cap INTR \neq \emptyset$; it is *valid in INTR* if $INTR \subseteq \|\varphi\|$. In this case, we also say that φ is a *validity of* DEL-PAO. A formula φ is *plainly satisfiable* if it has a model, i.e., if $\|\varphi\| \neq \emptyset$; it is *plainly valid* if φ is true in all models, i.e., if $\|\varphi\| = 2^{ATM}$. For example, $\mathtt{JS}\,p \wedge \neg\mathtt{S}_i\,p$ is plainly satisfiable but not satisfiable in *INTR*. On the other hand, $\neg[-\mathtt{S}_1\,\mathtt{S}_2\,p]\mathtt{JS}\,p$ is valid in *INTR* (and even plainly valid).

4.4. Relevant atoms

For atoms, the connection between visibility and knowledge is clear: the equivalences

$$K_i\alpha \leftrightarrow \mathtt{S}_i\,\alpha \wedge \alpha$$
$$K_i\neg\alpha \leftrightarrow \mathtt{S}_i\,\alpha \wedge \neg\alpha$$

are plainly valid. This motivates the notion of *relevant atoms* of a formula φ and of a program π, noted $RATM(\varphi)$ and $RATM(\pi)$, that is defined recursively as follows:

$$\begin{aligned}
RATM(\alpha) &= \{\alpha\}\\
RATM(\neg\varphi) &= RATM(\varphi)\\
RATM(\varphi\wedge\varphi') &= RATM(\varphi)\cup RATM(\varphi')\\
RATM(K_i\varphi) &= RATM(\varphi)\cup\{\mathtt{S}_i\,\alpha : \alpha\in RATM(\varphi)\}\\
RATM(CK\varphi) &= RATM(\varphi)\cup\{\mathtt{JS}\,\alpha : \alpha\in RATM(\varphi)\}\\
RATM([\pi]\varphi) &= RATM(\pi)\cup RATM(\varphi)\\
RATM(+\alpha) &= \{\alpha\}\\
RATM(-\alpha) &= \{\alpha\}\\
RATM(\pi;\pi') &= RATM(\pi)\cup RATM(\pi')\\
RATM(\pi\sqcup\pi') &= RATM(\pi)\cup RATM(\pi')\\
RATM(\varphi?) &= RATM(\varphi)
\end{aligned}$$

For example,

$$RATM(q\wedge CKK_ip) = \{q, p, \mathtt{S}_i\,p, \mathtt{JS}\,p, \mathtt{JS}\,\mathtt{S}_i\,p\}$$

This illustrates that $RATM(\varphi)$ includes $ATM(\varphi)$: we have $ATM(q\wedge CKK_ip) = \{q,p\}$.

In the gossip problem, we will mainly be interested in (conjunctions of) formulas of the form $K_{i_1}\dots K_{i_m}\alpha$. Here are two useful properties of such formulas.

Lemma 1. *The equivalence*

$$K_{i_1}\dots K_{i_m}\alpha \leftrightarrow \Big(\bigwedge_{\beta\in RATM(K_{i_1}\dots K_{i_m}\alpha)}\beta\Big)$$

is plainly valid, for $m \geq 0$.

Proof. We use induction on m. For $m = 0$ we have $RATM(\alpha) = \alpha$ and the equivalence is obvious. For $m \geq 1$, suppose

$$K_{i_1}\dots K_{i_m}\alpha \leftrightarrow \Big(\bigwedge_{\beta\in RATM(K_{i_1}\dots K_{i_m}\alpha)}\beta\Big)$$

is plainly valid. Take $V \in 2^{ATM}$. Then:

$$\begin{aligned}
&V \models K_iK_{i_1}\dots K_{i_m}\alpha\\
\text{iff } &V \models K_i\Big(\bigwedge_{\beta\in RATM(K_{i_1}\dots K_{i_m}\alpha)}\beta\Big)\\
\text{iff } &V \models \Big(\bigwedge_{\beta\in RATM(K_{i_1}\dots K_{i_m}\alpha)}K_i\beta\Big)\\
\text{iff } &V \models \Big(\bigwedge_{\beta\in RATM(K_{i_1}\dots K_{i_m}\alpha)}(\beta\wedge\mathtt{S}_i\,\beta)\Big)\\
\text{iff } &V \models \Big(\bigwedge_{\beta\in RATM(K_{i_1}\dots K_{i_m}\alpha)}\beta\Big)\wedge\Big(\bigwedge_{\beta\in RATM(K_{i_1}\dots K_{i_m}\alpha)}\mathtt{S}_i\,\beta\Big)\\
\text{iff } &V \models \Big(\bigwedge_{\beta\in RATM(K_{i_1}\dots K_{i_m}\alpha)}\beta\Big)\wedge\Big(\bigwedge_{\beta\in\{\mathtt{S}_i\,\gamma:\gamma\in RATM(K_{i_1}\dots K_{i_m}\alpha)\}}\beta\Big)\\
\text{iff } &V \models \Big(\bigwedge_{\substack{\beta\in RATM(K_{i_1}\dots K_{i_m}\alpha)\cup\\ \{\mathtt{S}_i\,\gamma:\gamma\in RATM(K_{i_1}\dots K_{i_m}\alpha)\}}}\beta\Big)\\
\text{iff } &V \models \Big(\bigwedge_{\beta\in RATM(K_iK_{i_1}\dots K_{i_m}\alpha)}\beta\Big),
\end{aligned}$$

hence the result. □

We write $\langle r_1,\dots,r_p\rangle \sqsubseteq \langle 1,\dots,m\rangle$ when $\langle r_1,\dots,r_p\rangle$ is a subsequence of $\{1,\dots,m\}$. For example, we have $\langle 1,3,4\rangle \sqsubseteq \langle 1,2,3,4,5\rangle$ and $\langle\rangle \sqsubseteq \langle 1,2,3,4,5\rangle$ but not $\langle 1,4,3\rangle \sqsubseteq \langle 1,2,3,4,5\rangle$.

Lemma 2. *We have, for $m \geq 0$:*

$$RATM(K_{i_1} \dots K_{i_m}\alpha) = \{\mathtt{S}_{i_{r_1}} \dots \mathtt{S}_{i_{r_p}}\alpha : \langle r_1, \dots, r_p\rangle \sqsubseteq \langle 1, \dots, m\rangle\}$$

Proof. We use induction on m. The case $m = 0$ is obvious since $RATM(\alpha) = \{\alpha\}$ and the only subsequence of $\langle\rangle$ is $\langle\rangle$. For $m \geq 1$, suppose

$$RATM(K_{i_1} \dots K_{i_m}\alpha) = \{\mathtt{S}_{i_{r_1}} \dots \mathtt{S}_{i_{r_p}}\alpha : \langle r_1, \dots, r_p\rangle \sqsubseteq \langle 1, \dots, m\rangle\}.$$

Then:

$$\begin{aligned}
&RATM(K_\ell K_{i_1} \dots K_{i_m}\alpha)\\
&= RATM(K_{i_1} \dots K_{i_m}\alpha) \cup \\
&\qquad \{\mathtt{S}_\ell\alpha' : \alpha' \in RATM(K_{i_1} \dots K_{i_m}\alpha)\}\\
&= \{\mathtt{S}_{i_{r_1}} \dots \mathtt{S}_{i_{r_p}}\alpha : \langle r_1, \dots, r_p\rangle \sqsubseteq \langle 1, \dots, m\rangle\} \cup \\
&\qquad \{\mathtt{S}_\ell\alpha' : \alpha' \in \{\mathtt{S}_{i_{r_1}} \dots \mathtt{S}_{i_{r_p}}\alpha :\\
&\qquad\qquad \langle r_1, \dots, r_p\rangle \sqsubseteq \langle 1, \dots, m\rangle\}\}\\
&= \{\mathtt{S}_{i_{r_1}} \dots \mathtt{S}_{i_{r_p}}\alpha : \langle r_1, \dots, r_p\rangle \sqsubseteq \langle 1, \dots, m\rangle\} \cup \\
&\qquad \{\mathtt{S}_\ell\mathtt{S}_{i_{r_1}} \dots \mathtt{S}_{i_{r_p}}\alpha : \langle r_1, \dots, r_p\rangle \sqsubseteq \langle 1, \dots, m\rangle\}\\
&= \{\mathtt{S}_{i_{r_1}} \dots \mathtt{S}_{i_{r_p}}\alpha : \langle r_1, \dots, r_p\rangle \sqsubseteq \langle \ell, 1, \dots, m\rangle\},
\end{aligned}$$

hence the result. □

Lemma 1 tells us that relevant atoms provide a link between visibility atoms—that are modified by the calls—and epistemic operators. Lemmas 1 and 2 will be useful in the next section when we will study properties of gossip calls.

5. Expressing calls in the language of DEL-PAO

Our logic provides the resources we need to model calls between agents and to reason about the evolution of their knowledge. Before proving that our algorithm is correct we show how to express calls and we establish their properties.

In the standard version of the gossip problem, agents only communicate their factual knowledge. As we have seen, they also have to tell what they know about others in order to achieve higher-order knowledge: for shared knowledge of level k they have to exchange all their knowledge up to depth $k-1$.

Let k be the level of shared knowledge to be attained. Let i and j be two agents. For a given integer m, we note $\{\mathtt{S}_i, \mathtt{S}_j\}^{\leq m}$ the set all non-empty sequences of visibility operators $\mathtt{S}_i$ and $\mathtt{S}_j$ of length at most m. For example:

$$\{\mathtt{S}_i, \mathtt{S}_j\}^{\leq 2} = \{\mathtt{S}_i, \mathtt{S}_j, \mathtt{S}_i\mathtt{S}_i, \mathtt{S}_i\mathtt{S}_j, \mathtt{S}_j\mathtt{S}_i, \mathtt{S}_j\mathtt{S}_j\}.$$

Then $Call^i_j$ is the sequential composition of programs of the form:

$$\begin{aligned}
&\textbf{if } K_iK_{i_1}\cdots K_{i_m}s \vee K_jK_{i_1}\cdots K_{i_m}s\\
&\qquad \textbf{then } +\sigma_1\mathtt{S}_{i_1}\cdots\mathtt{S}_{i_m}s; \dots; +\sigma_\ell\mathtt{S}_{i_1}\cdots\mathtt{S}_{i_m}s
\end{aligned}$$

for secret s in *Prop*, integer $m \leq k-1$, agents $i_1, \dots, i_m \in Agt$ and sequences $\{\mathtt{S}_i, \mathtt{S}_j\}^{\leq k-m} = \{\sigma_1, \dots, \sigma_\ell\}$. For example, for $k = 3$ the following is an element of the sequence:

$$\begin{aligned}
&\textbf{if } K_iK_\ell s \vee K_jK_\ell s\\
&\qquad \textbf{then } +\mathtt{S}_i\mathtt{S}_\ell s; +\mathtt{S}_j\mathtt{S}_\ell s; +\mathtt{S}_i\mathtt{S}_i\mathtt{S}_\ell s; +\mathtt{S}_i\mathtt{S}_j\mathtt{S}_\ell s;\\
&\qquad\qquad +\mathtt{S}_j\mathtt{S}_i\mathtt{S}_\ell s; +\mathtt{S}_j\mathtt{S}_j\mathtt{S}_\ell s
\end{aligned}$$

That piece of program tests whether $K_\ell s$ is known by i or j and if so makes $\mathtt{S}_\ell s$ visible for both i and j and i's observation of $\mathtt{S}_\ell s$ visible for j, and vice versa. (Observe that the additions $+\mathtt{S}_i\mathtt{S}_i\mathtt{S}_k s$ and $+\mathtt{S}_j\mathtt{S}_j\mathtt{S}_k s$ are trivial because they are introspectively valid.)

Some properties of the program $Call^i_j$ and its interaction with the shared knowledge operator via the relevant atoms will be useful in our proofs.

First of all, the dynamic operators $[Call^i_j]$ and the shared knowledge operators EK_J are normal modal operators. So in particular

$$[Call^i_j]\varphi \wedge [Call^i_j]\psi \leftrightarrow [Call^i_j](\varphi \wedge \psi)$$

and

$$(EK_J\varphi \wedge EK_J\psi) \leftrightarrow EK_J(\varphi \wedge \psi)$$

are plainly valid. Moreover, we can put coalitions together: the schema

$$(EK_{J_1}\varphi \wedge EK_{J_2}\varphi) \leftrightarrow EK_{J_1\cup J_2}\varphi$$

is plainly valid for every $J_1, J_2 \subseteq Agt$. (To see this reduce EK according to its definition.) Finally, calls preserve positive knowledge and produce shared knowledge, which is a property that we state formally:

Proposition 2. *Let $s \in \{s_i : i \in Agt\}$ and $m \geq 0$. Let φ be of the form either $K_{i_1} \dots K_{i_m}s$ or $EK_{J_1} \dots EK_{J_m}s$. Then the formulas:*

$$\varphi \to [Call^i_j]\varphi \qquad (Prsv)$$
$$K_i\varphi \to [Call^i_j]EK^{k-m}_{i,j}\varphi \qquad (Incr)$$

are plainly valid.

Proof. We prove each implication thanks to properties of DEL-PAO.

The proof of (*Prsv*) is straightforward: it is plainly valid because φ does not contain negations and calls only make atoms true.

The proof of (*Incr*) is a bit more involved. We only prove the case where $\varphi = K_{i_1} \dots K_{i_m} s$; the case $\varphi = EK_{J_1} \dots EK_{J_m} s$ is similar since EK_J is a conjunction of K_i. We have seen in Lemma 1 that

$$\varphi \leftrightarrow \Big(\bigwedge_{\beta \in RATM(\varphi)} \beta \Big)$$

is plainly valid. Moreover, we have:

$$RATM(EK^m_{i,j}\varphi) = \{\sigma\alpha : \alpha \in RATM(\varphi), \sigma \in \{\mathsf{s}_i, \mathsf{s}_j\}^{\leq m}\}$$

by the definition of $RATM(.)$. Therefore we want to prove that:

$$K_i\varphi \to [Call^i_j]\Big(\bigwedge_{\beta \in \{\sigma\alpha : \alpha \in RATM(\varphi), \sigma \in \{\mathsf{s}_i, \mathsf{s}_j\}^{\leq k-m}\}} \beta \Big).$$

We have:

$$K_iK_{i_1} \dots K_{i_m} s \to K_iK_{i_{r_1}} \dots K_{i_{r_p}} s,$$

for every $\langle r_1, \dots, r_p \rangle \sqsubseteq \langle 1, \dots, m \rangle$, by axiom T of standard epistemic logic. Since $K_iK_{i_{r_1}} \dots K_{i_{r_p}} s$ obviously implies $K_iK_{i_{r_1}} \dots K_{i_{r_p}} s \vee K_jK_{i_{r_1}} \dots K_{i_{r_p}} s$, for every subset $\langle r_1, \dots, r_p \rangle \sqsubseteq \langle 1, \dots, m \rangle$, we have:

$$K_iK_{i_1} \dots K_{i_m} s \to [Call^i_j]\Big(\bigwedge_{\beta \in \{\sigma \mathsf{s}_{i_{r_1}} \dots \mathsf{s}_{i_{r_p}} s : \sigma \in \{\mathsf{s}_i, \mathsf{s}_j\}^{\leq k-p}\}} \beta \Big),$$

by the definition of the programs composing $Call^i_j$. This means that:

$$K_iK_{i_1} \dots K_{i_m} s \to [Call^i_j]\Big(\bigwedge_{\beta \in \{\sigma \mathsf{s}_{i_{r_1}} \dots \mathsf{s}_{i_{r_p}} s : \sigma \in \{\mathsf{s}_i, \mathsf{s}_j\}^{\leq k-m}\}} \beta \Big),$$

because $p \leq m$. Since this is true for any $\langle r_1, \dots, r_p \rangle \sqsubseteq \langle 1, \dots, m \rangle$, we can apply Lemma 2 and obtain:

$$K_iK_{i_1} \dots K_{i_m} s \to [Call^i_j]\Big(\bigwedge_{\beta \in \{\sigma\alpha : \alpha \in RATM(K_{i_1} \dots K_{i_m} s), \sigma \in \{\mathsf{s}_i, \mathsf{s}_j\}^{\leq k-m}\}} \beta \Big),$$

which is our result. □

Formally, the program corresponding to the turn t of Algorithm 1 is therefore:

$$\mathsf{turn}_t = Call^{left}_{n-2-t}; \dots; Call^{left}_{n-3}; Call^{left}_0; \dots; Call^{left}_{n-4-t}; Call^{right}_{n-3-t}.$$

6. Correctness of the algorithm

We now prove that the algorithm returns a solution. Let us write the set of agents as

$$Agt = \{left, right, 0, \dots, n-3\}.$$

Remember that $Prop = \{s_i : i \in Agt\}$ is the set of propositional variables. The initial state is modelled by the valuation

$$V_0 = \{s_i : i \in Agt\} \cup \{\mathsf{s}_i s_i : i \in Agt\} \cup \{\alpha : \alpha \text{ is valid in } INTR\}.$$

So all secrets are true, each agent knows her own secret, and moreover the introspectively valid atoms are true (so V_0 is introspective). We have:

$$V_0 \models \bigwedge_{i \in Agt} \Big(K_i s_i \wedge \bigwedge_{j \in Agt, j \neq i} \neg K_j s_i \Big).$$

An agent is an *expert for depth t* if her personal goal for depth t is reached. Precisely, agent i is an expert for depth $t \geq 1$ if and only if we have:

$$K_i EK^{t-1}_{Agt} s_{Agt}.$$

The dynamic modalities of DEL-PAO nicely allow to express that a further call would turn an agent i into an expert, i.e., that i is a semi-expert. Indeed, two agents i and j are *complementary for depth t* ('semi-experts'), noted $\mathsf{compl}_t(i, j)$, if a call between i and j would make them both experts for depth t. More formally:

$$\mathsf{compl}_t(i, j) \stackrel{\text{def}}{=} [Call^i_j] EK_{i,j} EK^{t-1}_{Agt} s_{Agt}.$$

Furthermore, two pairs of agents (i_1, i_2) and (j_1, j_2) are complementary for depth t if and only if we have:

$$\mathsf{compl}_t(i_1, j_1) \wedge \mathsf{compl}_t(i_1, j_2) \wedge \mathsf{compl}_t(i_2, j_1) \wedge \mathsf{compl}_t(i_2, j_2).$$

We will prove that at each turn, two pairs of agents are complementary: the first pair is agent *left* along with the last agent she called at this turn, and the second is agent *right* along with the last (and only agent) she called at this turn.

The first turn is a special case where semi-experts of depth 1 are produced.

Lemma 3. *We have:*

$$V_0 \models [\mathsf{turn}_0]\big(EK_{left,n-4}s_{left,0,\ldots,n-4} \wedge EK_{right,n-3}s_{right,n-3}\big).$$

Proof. Let us simply write "ij" for the call between i and j in proofs. The first turn (turn 0) of Algorithm 1 produces the following sequence of calls:

$$left0; left1; \ldots; left(n-4); right(n-3).$$

By formula (*Incr*) of Proposition 2 we have:

$$V_0 \models [Call_0^{left}]EK_{left,0}s_{left,0}$$

and therefore:

$$V_0 \models [Call_0^{left}]K_{left}s_{left,0}.$$

We do the same for the next call:

$$\begin{aligned} V_0 &\models [Call_0^{left}; Call_1^{left}]EK_{left,1}s_{left,0,1} \\ \Rightarrow V_0 &\models [Call_0^{left}; Call_1^{left}]K_{left}s_{left,0,1}. \end{aligned}$$

And so on until:

$$V_0 \models [Call_0^{left}; Call_1^{left}; \ldots; Call_{n-4}^{left}] EK_{left,n-4}s_{left,0,1,\ldots,n-4}.$$

In the same vein we also have:

$$V_0 \models [Call_{n-3}^{right}]EK_{right,n-3}s_{right,n-3}.$$

By (*Prsv*) of Proposition 2 we then obtain:

$$V_0 \models [Call_0^{left}; \ldots; Call_{n-4}^{left}; Call_{n-3}^{right}] \big(EK_{left,n-4}s_{left,0,\ldots,n-4} \wedge EK_{right,n-3}s_{right,n-3}\big)$$

which is the same as:

$$V_0 \models [\mathsf{turn}_0]\big(EK_{left,n-4}s_{left,0,\ldots,n-4} \wedge EK_{right,n-3}s_{right,n-3}\big),$$

hence the result. □

We now characterize the turns after turn_0.

Lemma 4. *For $t \geq 1$, we have:*

$$V_0 \models [\mathsf{turn}_0; \ldots; \mathsf{turn}_t] \big(EK_{left,n-4-t}EK_{left,0-t,\ldots,n-4-t}EK_{Agt}^{t-1}s_{Agt} \wedge EK_{right,n-3-t}EK_{right,n-3-t}EK_{Agt}^{t-1}s_{Agt}\big).$$

Proof. We use induction on t. Both cases resemble the proof of Lemma 3.

Base case: $t = 1$. Turn 1 of Algorithm 1 produces the following sequence:

$$left(n-3); left0; left1; \ldots; left(n-5); right(n-4).$$

By Lemma 3 and (*Incr*) of Proposition 2 we have:

$$\begin{aligned} V_0 &\models [\mathsf{turn}_0][Call_{n-3}^{left}]EK_{left,n-3}EK_{left,n-3}s_{Agt} \\ \Rightarrow V_0 &\models [\mathsf{turn}_0][Call_{n-3}^{left}]K_{left}EK_{left,n-3}s_{Agt}. \end{aligned}$$

Then again by (*Incr*):

$$\begin{aligned} V_0 &\models [\mathsf{turn}_0][Call_{n-3}^{left}; Call_0^{left}]EK_{left,0}EK_{left,n-3,0}s_{Agt} \\ \Rightarrow V_0 &\models [\mathsf{turn}_0][Call_{n-3}^{left}; Call_0^{left}]K_{left}EK_{left,n-3,0}s_{Agt}, \end{aligned}$$

and for the next call:

$$\begin{aligned} V_0 &\models [\mathsf{turn}_0][Call_{n-3}^{left}; Call_0^{left}; Call_1^{left}] EK_{left,1}EK_{left,n-3,0,1}s_{Agt} \\ \Rightarrow V_0 &\models [\mathsf{turn}_0][Call_{n-3}^{left}; Call_0^{left}; Call_1^{left}] K_{left}EK_{left,n-3,0,1}s_{Agt}, \end{aligned}$$

and so on until:

$$V_0 \models [\mathsf{turn}_0][Call_{n-3}^{left}; Call_0^{left}; Call_1^{left}; \ldots; Call_{n-5}^{left}] EK_{left,n-5}EK_{left,n-3,0,1,\ldots,n-5}s_{Agt}.$$

Similarly we have:

$$V_0 \models [\mathsf{turn}_0][Call_{n-4}^{right}]EK_{right,n-4}EK_{right,n-4}s_{Agt}.$$

Finally we obtain the result by (*Prsv*) of Proposition 2:

$$V_0 \models [\mathsf{turn}_0][Call^{left}_{n-3}; Call^{left}_{0}; \ldots; Call^{left}_{n-5}; Call^{right}_{n-4}] \big(EK_{left,n-5}EK_{left,n-3,0,1,\ldots,n-5}s_{Agt} \wedge EK_{right,n-4}EK_{right,n-4}s_{Agt}\big),$$

that is:

$$V_0 \models [\mathsf{turn}_0;\mathsf{turn}_1] \big(EK_{left,n-5}EK_{left,n-3,0,1,\ldots,n-5}s_{Agt} \wedge EK_{right,n-4}EK_{right,n-4}s_{Agt}\big).$$

Inductive case. The reasoning is similar, but generalized to turn $t+1$. Suppose the formula is true for turn t. The turn $t+1$ is:

$$left(n-3-t); left(0-t); \ldots; left(n-5-t); right(n-4-t).$$

By our induction hypothesis and (*Incr*) of Proposition 2 we have:

$$V_0 \models [\mathsf{turn}_0;\ldots;\mathsf{turn}_t][Call^{left}_{n-3-t}] EK_{left,n-3-t}EK_{left,n-3-t}EK_{Agt}EK^{t-1}_{Agt}s_{Agt},$$

that is:

$$V_0 \models [\mathsf{turn}_0;\ldots;\mathsf{turn}_t][Call^{left}_{n-3-t}] EK_{left,n-3-t}EK_{left,n-3-t}EK^{t}_{Agt}s_{Agt},$$

which implies:

$$V_0 \models [\mathsf{turn}_0;\ldots;\mathsf{turn}_t][Call^{left}_{n-3-t}] K_{left}EK_{left,n-3-t}EK^{t}_{Agt}s_{Agt}.$$

Then by (*Prsv*) of Proposition 2:

$$\begin{aligned} V_0 &\models [\mathsf{turn}_0;\ldots;\mathsf{turn}_t][Call^{left}_{n-3-t}; Call^{left}_{0-t}] EK_{left,0-t}EK_{left,n-3-t,0-t}EK^{t}_{Agt}s_{Agt} \\ \Rightarrow V_0 &\models [\mathsf{turn}_0;\ldots;\mathsf{turn}_t][Call^{left}_{n-3-t}; Call^{left}_{0-t}] K_{left}EK_{left,n-3-t,0-t}EK^{t}_{Agt}s_{Agt}, \end{aligned}$$

…and so on until:

$$V_0 \models [\mathsf{turn}_0;\ldots;\mathsf{turn}_t][Call^{left}_{n-3-t}; Call^{left}_{0-t}; \ldots; Call^{left}_{n-5-t}] EK_{left,n-5-t}EK_{left,n-3-t,0-t,\ldots,n-5-t}EK^{t}_{Agt}s_{Agt}.$$

Moreover, by (*Incr*):

$$V_0 \models [\mathsf{turn}_0;\ldots;\mathsf{turn}_t][Call^{right}_{n-4-t}] EK_{right,n-4-t}EK_{right,n-4-t}EK_{Agt}EK^{t-1}_{Agt}s_{Agt},$$

that is:

$$V_0 \models [\mathsf{turn}_0;\ldots;\mathsf{turn}_t][Call^{right}_{n-4-t}] EK_{right,n-4-t}EK_{right,n-4-t}EK^{t}_{Agt}s_{Agt}.$$

We end as usual with (*Prsv*):

$$\begin{aligned} V_0 &\models [\mathsf{turn}_0;\ldots;\mathsf{turn}_t][Call^{left}_{n-3-t};\ldots; Call^{left}_{n-5-t}; Call^{right}_{n-4-t}] \big(EK_{left,n-5-t}EK_{left,n-3-t,\ldots,n-5-t}EK^{t}_{Agt}s_{Agt} \wedge EK_{right,n-4-t}EK_{right,n-4-t}EK^{t}_{Agt}s_{Agt}\big) \\ \Leftrightarrow V_0 &\models [\mathsf{turn}_0;\ldots;\mathsf{turn}_t;\mathsf{turn}_{t+1}] \big(EK_{left,n-5-t}EK_{left,n-3-t,\ldots,n-5-t}EK^{t}_{Agt}s_{Agt} \wedge EK_{right,n-4-t}EK_{right,n-4-t}EK^{t}_{Agt}s_{Agt}\big), \end{aligned}$$

which is our result for $t+1$. □

Lemma 5. *After turn $t-1$ of Algorithm 1, the pairs $(left, n-3-t)$ and $(right, 0-t)$ are complementary for depth t.*

Proof. From Lemma 4 we can deduce:

$$V_0 \models [\mathsf{turn}_0;\ldots;\mathsf{turn}_{t-1}] \big(K_{left}EK_{left,1-t,\ldots,n-3-t}EK^{t-2}_{Agt}s_{Agt} \wedge K_{right}EK_{right,0-t}EK^{t-2}_{Agt}s_{Agt}\big).$$

Applying (*Incr*) of Proposition 2 we obtain:

$$V_0 \models [\mathsf{turn}_0;\ldots;\mathsf{turn}_{t-1}][Call^{left}_{right}] EK_{left,right}EK_{Agt}EK^{t-2}_{Agt}s_{Agt},$$

that is:

$$V_0 \models [\mathsf{turn}_0;\ldots;\mathsf{turn}_{t-1}][Call^{left}_{right}] EK_{left,right}EK^{t-1}_{Agt}s_{Agt},$$

which is equivalent to:

$$V_0 \models [\mathsf{turn}_0;\ldots;\mathsf{turn}_{t-1}]\mathsf{compl}_t(left, right).$$

By the same reasoning for *left* and $0-t$, *right* and $n-3-t$, and finally $n-3-t$ and $0-t$, we obtain that each of them are complementary, hence the result. □

Lemma 6. *The goal for depth t, $EK^t_{Agt}s_{Agt}$, is reached after turn t of Algorithm 1.*

Proof. Turn t of Algorithm 1 is:

$$\mathit{left}(0{-}t), \mathit{left}(1{-}t), \ldots, \mathit{left}(n{-}4{-}t), \mathit{right}(n{-}3{-}t).$$

By Lemma 5, after turn $t{-}1$ and the first call $\mathit{left}(0{-}t)$ of turn t, agents *left* and $0{-}t$ become experts for depth t. (Thus $EK_{\mathit{left},0-t}EK^{t-1}_{Agt}s_{Agt}$.) Then after the $n{-}4$ calls $\mathit{left}(1{-}t), \ldots, \mathit{left}(n{-}4{-}t)$ we get by (*Incr*) of Proposition 2:

$$K_{1-t}EK^{t-1}_{Agt}s_{Agt} \wedge \ldots \wedge K_{n-4-t}EK^{t-1}_{Agt}s_{Agt},$$

that is, $1{-}t$, …, $n{-}4{-}t$ are all experts for depth t. Finally, after the last call $\mathit{right}(n{-}3{-}t)$, and also by Lemma 5, agents *right* and $n{-}3{-}t$ become experts for depth t. (Thus $EK_{\mathit{right},n-3-t}EK^{t-1}_{Agt}s_{Agt}$.) Hence after the $n{-}2$ calls of turn t we have $EK_{Agt}EK^{t-1}_{Agt}s_{Agt}$, which is equivalent to $EK^t_{Agt}s_{Agt}$. □

Proposition 3. *The sequence resulting from Algorithm 1 gives a solution to the generalized gossip problem for $k{\geq}1$ and $n{\geq}4$.*

Proof. By Lemma 6, the goal for depth t is reached after turn t of Algorithm 1. Thus the goal for depth k is reached after turn k ($k{+}1$ turns), i.e., at the end of the algorithm. □

Proposition 3 implies that at most $(k{+}1)(n{-}2)$ calls are required to solve the instance $\mathit{Gossip}(k,n)$ of the generalized gossip problem.

7. Optimality of the algorithm

In this section, we prove that the sequence of calls returned by our algorithm has an optimal length. Our optimality result is derived from that of [18].

We first prove a property of the gossip problem that may seem obvious but that we prefer to clarify.

Lemma 7. *Suppose m agents know a fact φ not known to the remaining agents. Then it takes at least $n{-}m$ calls for the remaining agents to learn φ.*

Proof. It suffices to prove that a call Call^i_j increases the knowledge on a fact φ of at most one agent. We distinguish four cases, depending on the knowledge of i and j about φ.

- **Neither i nor j know φ.** Then $K_i\varphi \vee K_j\varphi$ is false and no agent knows φ after Call^i_j.
- **i knows φ but j does not know φ.** Then $K_i\varphi \vee K_j\varphi$ is true and both agents know φ after Call^i_j, but only j learned it.
- **i does not know φ but j knows φ.** Then $K_i\varphi \vee K_j\varphi$ is true and both agents know φ after Call^i_j, but only i learned it.
- **i and j know φ.** Then $K_i\varphi \vee K_j\varphi$ is true and both agents know φ after Call^i_j, but no one learned it.

Therefore $n{-}m$ calls are necessary to spread a piece of gossip to $n{-}m$ agents. □

Proposition 4. *The minimal number of calls needed to solve the instance $\mathit{Gossip}(k,n)$ of the generalized gossip problem, for $k{\geq}1$ and $n{\geq}4$, is at least $(k{+}1)(n{-}2)$.*

Proof. We use induction on k.

Base case: $k{=}1$. As we have seen, for $k{=}1$ the lower bound $2(n{-}2)$ was established in the literature [8,32, 25].

Inductive case. Suppose that at least $(k{+}1)(n{-}2)$ calls are needed to achieve the goal for depth k. This implies that after $(k{+}1)(n{-}2)-1$ calls, at least one agent, let us call her i, does not know a piece of information of depth $k{-}1$:

$$V_0 \models [\mathit{Call}^{i_1}_{j_1}; \ldots; \mathit{Call}^{i_{(k+1)(n-2)-1}}_{j_{(k+1)(n-2)-1}}] \neg K_i K_{\ell_1} \cdots K_{\ell_{k-1}} s.$$

Remark that i could also ignore facts of a lower depth; in this case she would also ignore facts of depth $k{-}1$ by the truth axiom T. For example, suppose that i does not know the secret of 1; then she cannot know that 2 knows the secret of 1, and that 1 knows that 2 knows the secret of 1, and so on.

Then the $(k{+}1)(n{-}2)$-th call involves i (otherwise her knowledge does not evolve) and another agent, say j. It establishes not only $K_iK_{\ell_1} \cdots K_{\ell_{k-1}}s$ and thus the goal for depth k:

$$V_0 \models \langle \mathit{Call}^{i_1}_{j_1}; \ldots; \mathit{Call}^{i_{(k+1)(n-2)-1}}_{j_{(k+1)(n-2)-1}} \rangle \langle \mathit{Call}^i_j \rangle EK^k_{Agt}s_{Agt},$$

but also the fact that i and j both know this:

$$V_0 \models \langle \mathit{Call}^{i_1}_{j_1}; \ldots; \mathit{Call}^{i_{(k+1)(n-2)-1}}_{j_{(k+1)(n-2)-1}} \rangle \langle \mathit{Call}^i_j \rangle EK_{i,j}EK^k_{Agt}s_{Agt},$$

while no other agent does:

$$V_0 \models [Call^{i}_{j_1};\ldots;Call^{i_{(k+1)(n-2)-1}}_{j_{(k+1)(n-2)-1}}][Call^{i}_{j}] \Big(\bigwedge_{\ell \in Agt\setminus\{i,j\}} \neg K_\ell EK^k_{Agt} s_{Agt} \Big).$$

To establish the goal for depth k+1, i.e., to establish $EK_{Agt}EK^k_{Agt}s_{Agt}$, it is necessary to distribute $EK^k_{Agt}s_{Agt}$ from i and j to all other agents. By Lemma 7, we know that this takes at least $n-2$ calls. Therefore, we need $(k+1)(n-2)+n-2=(k+2)(n-2)$ calls to achieve the goal for depth k+1. □

Propositions 3 and 4 together ensure Theorem 1, i.e., that the minimal number of calls needed to solve the instance $Gossip(k,n)$ of the generalized gossip problem, for $k \geq 1$ and $n \geq 4$, is exactly $(k+1)(n-2)$.

8. The case of two and three agents

Our algorithm only works when four or more agents are involved; it cannot be applied when there are only two and three agents.

The former case is easy: only one call is necessary for two agents to reach knowledge on their secrets of level k, whatever k is. (This is ensured by formula (*Incr*) of Proposition 2.) Obviously, less calls are not sufficient given the agents' initial ignorance about the other's secret.

Consider the case of three agents, say 0, 1 and 2. We give an algorithm that takes k+2 turns of one call each.

Algorithm 2. *For* $t = 0..k$ *do*

agent 0 *calls agent* $(t \pmod 2)+1$.

The algorithm consists in sequences of the form $Call^0_1;Call^0_2;Call^0_1;Call^0_2;\ldots$, k+2 times. Hence each turn contains one call: $\mathsf{turn}_0 = Call^0_1$, $\mathsf{turn}_1 = Call^0_2$, $\mathsf{turn}_2 = Call^0_1$, and so on. We prove that the algorithm is correct and optimal.

Theorem 2. *The minimal number of calls to solve the instance* $Gossip(k,3)$ *of the generalized gossip problem, for* $k \geq 1$*, is* k+2.

Proposition 5. *The sequence resulting from Algorithm 2 gives a solution to the generalized gossip problem for* $k \geq 1$ *and* n=3.

Proof. We use induction on k.

Base case: k=1. It is easy to check that the sequence $Call^0_1;Call^0_2;Call^0_1$ establishes the goal for depth 1.

Inductive case. Suppose k+2 turns of the algorithm achieve the goal for depth k:

$$V_0 \models [\mathsf{turn}_0;\mathsf{turn}_1;\ldots;\mathsf{turn}_{k+2}]EK^k_{Agt}s_{Agt}.$$

Because $\mathsf{turn}_{k+2} = Call^0_{(k+2 \pmod 2)+1}$, we also have:

$$V_0 \models [\mathsf{turn}_0;\mathsf{turn}_1;\ldots;\mathsf{turn}_{k+2}] EK_{0,(k+2 \pmod 2)+1}EK^k_{Agt}s_{Agt},$$

which implies:

$$V_0 \models [\mathsf{turn}_0;\mathsf{turn}_1;\ldots;\mathsf{turn}_{k+2}] K_0EK^k_{Agt}s_{Agt},$$

and hence by (*Incr*) of Proposition 2:

$$V_0 \models [\mathsf{turn}_0;\mathsf{turn}_1;\ldots;\mathsf{turn}_{k+2}][\mathsf{turn}_{k+3}] EK_{0,(k+3 \pmod 2)+1}EK^k_{Agt}s_{Agt}.$$

since $\mathsf{turn}_{k+3} = Call^0_{(k+3 \pmod 2)+1}$. Therefore, because 0, $(k+2 \pmod 2)+1$ and $k+3 \pmod 2)+1$ are all different, we obtain by formula (*Prsv*) of Proposition 2:

$$V_0 \models [\mathsf{turn}_0;\mathsf{turn}_1;\ldots;\mathsf{turn}_{k+2}][\mathsf{turn}_{k+3}] EK_{Agt}EK^k_{Agt}s_{Agt},$$

that is:

$$V_0 \models [\mathsf{turn}_0;\mathsf{turn}_1;\ldots;\mathsf{turn}_{k+2};\mathsf{turn}_{k+3}] EK^{k+1}_{Agt}s_{Agt},$$

which is our goal for depth k+1. □

Proposition 6. *The minimal number of calls needed to solve the instance* $Gossip(k,3)$ *of the generalized gossip problem, for* $k \geq 1$*, is at least* k+2.

Proof. This proof is similar to the proof of optimality for $n \geq 4$ (see Proposition 4).

Base case: k=1. It was proven—and it is easy to check—that 3 calls are necessary, in the original problem, when three agents are involved [8,32,25].

Inductive case. Suppose that at least $k+2$ calls are necessary to achieve the goal for depth k. Then after $k+1$ calls, at least one agent i does not know a piece of information of depth $k-1$. The $k+2$-th call, between i and j, makes i know this piece of information, and i and j be aware of this. At least one call is necessary to inform the third agent that the goal for depth k was reached, establishing the goal for depth $k+1$ in $k+3$ calls. □

This establishes Theorem 2, i.e., the minimal number of calls needed to solve the instance $Gossip(k,3)$ of the generalized gossip problem is exactly $k+2$.

9. Gossiping with ignorance goals

In the gossip problem, we aim at full knowledge of every agent. We could also consider scenarios where we want that some agents do not learn some secrets. Assuming agents are obliged to tell all the secrets they know (say, for example, because they do not know the goal), the ordering of calls might be influenced by these 'ignorance goals.' This section discusses some aspects of this variant which, to the best of our knowledge, was not investigated before. Unlike the version with the 'full knowledge' goal, we do not provide a result on the number of calls or a generic algorithm, but rather protocols for some specific cases and general remarks.

Let us start with the original gossip problem where the goal is to achieve shared knowledge of order $k=1$. Suppose we do not want agent 1 to know the secret of 2, and full knowledge otherwise. Our goal is:

$$\big(EK_{Agt\setminus\{1\}}s_{Agt}\big) \wedge \big(K_1 s_{Agt\setminus\{2\}} \wedge \neg K_1 s_2\big).$$

While this is obviously unsolvable for 2 agents, for at least 3 agents a successful protocol is one where 1 calls every other agent but 2, and then all agents but 1 call each other until their knowledge is shared. Slightly more generally, if 1 must not know the secret of 2, …, m then she should call $m+1$, …, n before they call any of 2, …, m, then 2, …, m, $m+1$, …, n can freely share their knowledge.

Things get quickly complicated, even for $k=1$, when we require several agents to be ignorant. For example, suppose $n=4$ and suppose agent 1 should not know the secret of agent 3, while 2 should not know the secret of 4. Then 1 should call 2 first, before she calls 4 and before 2 calls 3. Then 1 can call 4, 2 can call 3 and 3 can call 4. Now suppose that 1 should not know the secret of 3, while 3 should not know the secret of 1. Then no sequence of calls leads to a solution, since every agent that 1 calls cannot be called by 3 and conversely.

Now consider higher-order order goals. Take $k=2$ and suppose we want 1 not to know whether 2 knows the secret of 3 (but we do want 2 to know the secret of 3). This means that the goal is:

$$\big(EK_{Agt\setminus\{1\}}EK_{Agt}s_{Agt}\big) \wedge \big((K_1 EK_{Agt\setminus\{2\}}s_{Agt} \wedge K_1K_2 s_{Agt\setminus\{3\}}) \wedge \neg K_1K_2s_3\big).$$

Then the following protocol is a solution:

1. Agent 2 calls every agent but 3;
2. All agents but 2 call each other until full knowledge of depth 2 is acquired;
3. Agent 2 calls every agent but 1.

After the second step, every agent i different from 2 has almost reached her goal, except that she does not know whether 2 knows the secret of 3 (because it is not the case yet). At the third step, 2 calls everyone but 1 in order to learn 3's secret, acquire the required depth of knowledge and inform every other agent.

Remark that if we increase the depth k, the goal that *only* 1 does not know whether 2 knows the secret of 3 becomes unsolvable. Indeed, by the truth axiom T of epistemic logic, we have, for example:

$$K_1K_4K_2s_3 \rightarrow K_1K_2s_3,$$

and hence the latter cannot be false without the former being false. Therefore for $k>2$, the correct specification will be that every goal of the form:

$$K_1K_{i_1}\ldots K_{i_m}K_2K_{j_1}\ldots K_{j_p}s_3$$

is false, for $m+p+2 \leq k$.

10. Conclusion

We have provided a logical analysis of the gossip problem, focusing on how higher-order shared knowledge can be obtained. We did so in a particular dynamic epistemic logic: Dynamic Epistemic Logic of Propositional Assignment and Observation DEL-PAO. Its integration of knowledge modalities and dynamic modalities provides a handy language in order to reason about concepts such as an agent being a semi-expert, which is pivotal in our algorithm. With DEL-

PAO, we were able to prove both the correctness and optimality of our algorithm, generalizing the results from [8,32,25] on the minimal number of calls.

The gossip problem recently attracted quite some attention in the dynamic epistemic logic community [4,5,38]. The version that we have studied here has a central scheduler telling each agent when to act and what to do. Other versions where the protocol is distributed were investigated recently [3,19]. We believe that our generalization—as well as further variations where e.g. calls can only be made according to some graph structure—provide interesting, canonical multi-agent planning problems that can be compared to the blocksworld in classical planning. This is the subject of ongoing work; first steps are reported in [17,30]. We believe that our visibility-based approach is an interesting alternative to dynamic epistemic logic-based planning that was proposed in [13]: indeed, it was shown that such planning problems are undecidable [7], and so even for simple instances [15].

Acknowledgements

We would like to thank the reviewers of AT'2015 for their comments which hopefully helped to improve the presentation of the paper. We also acknowledge several discussions about the gossip problem at the inspiring August 2015 Lorentz Center workshop "To be announced" in Leiden, in particular with Hans van Ditmarsch, Jan van Eijck, Malvin Gattinger, Louwe Kuijer, Christian Muise, Pere Pardo, Rahim Ramezanian and Francois Schwarzentruber. We are also grateful to Thomas Bolander, Davide Grossi, Emiliano Lorini, Frédéric Maris, Martin Cooper, and Pierre Regnier for many discussions about the gossip problem.

References

[1] E. A. Akkoyunlu, K. Ekanadham, and R. V. Hubert. Some constraints and tradeoffs in the design of network communications. In *Proceedings of the 5th ACM Symposium on Operating Systems Principles*, pages 67–74. ACM Press, 1975.

[2] R. Alur, T. A. Henzinger, F. Y. C. Mang, S. Qadeer, S. K. Rajamani, and S. Tasiran. MOCHA: modularity in model checking. In A. J. Hu and M. Y. Vardi, editors, *Computer Aided Verification, 10th International Conference, CAV '98, Vancouver, BC, Canada, June 28 - July 2, 1998, Proceedings*, volume 1427 of *Lecture Notes in Computer Science*, pages 521–525. Springer, 1998.

[3] K. R. Apt, D. Grossi, and W. van der Hoek. Epistemic protocols for distributed gossiping. In R. Ramanujam, editor, *Proceedings Fifteenth Conference on Theoretical Aspects of Rationality and Knowledge, TARK 2015, Carnegie Mellon University, Pittsburgh, USA, June 4-6, 2015.*, volume 215 of *EPTCS*, pages 51–66, 2015.

[4] M. Attamah, H. van Ditmarsch, D. Grossi, and W. van der Hoek. A framework for epistemic gossip protocols. In N. Bulling, editor, *Multi-Agent Systems - 12th European Conference, EUMAS 2014, Prague, Czech Republic, December 18-19, 2014, Revised Selected Papers*, volume 8953 of *Lecture Notes in Computer Science*, pages 193–209. Springer, 2014.

[5] M. Attamah, H. van Ditmarsch, D. Grossi, and W. van der Hoek. Knowledge and gossip. *Proceedings of 21st ECAI*, pages 21–26, 2014.

[6] M. Attamah, H. van Ditmarsch, D. Grossi, and W. van der Hoek. The pleasure of gossip. In L. M. C. Baskent and R. Ramanujam, editors, *Rohit Parikh on Logic, Language and Society*. Springer, to appear.

[7] G. Aucher and T. Bolander. Undecidability in epistemic planning. In F. Rossi, editor, *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 27–33. IJCAI/AAAI, 2013.

[8] B. Baker and R. Shostak. Gossips and telephones. *Discrete Mathematics*, 2(3):191–193, 1972.

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

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

[11] A. Baltag, L. S. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. In *Proc. TARK'98*, pages 43–56. Morgan Kaufmann, 1998.

[12] T. Bolander. Seeing is believing: Formalising false-belief tasks in dynamic epistemic logic. In A. Herzig and E. Lorini, editors, *Proceedings of the European Conference on Social Intelligence (ECSI-2014), Barcelona, Spain, November 3-5, 2014.*, volume 1283 of *CEUR Workshop Proceedings*, pages 87–107. CEUR-WS.org, 2014.

[13] T. Bolander and M. B. Andersen. Epistemic planning for single and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21(1):9–34, 2011.

[14] T. Charrier, A. Herzig, E. Lorini, and F. Schwarzentruber. Building epistemic logic from observations and public announcements. In *International Conference on Principles of Knowledge Representation and Reasoning (KR), Cape Town*, pages 268–277, http://www.aaai.org/Press/press.php, 2016. AAAI Press.

[15] T. Charrier, B. Maubert, and F. Schwarzentruber. On the impact of modal depth in epistemic planning. In Kambhampati [29], pages 1030–1036.

[16] T. Charrier and F. Schwarzentruber. Arbitrary public announcement logic with mental programs. In G. Weiss, P. Yolum, R. H. Bordini, and E. Elkind, editors, *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pages 1471–1479. ACM, 2015.

[17] M. C. Cooper, A. Herzig, F. Maffre, F. Maris, and P. Régnier. A simple account of multi-agent epistemic planning. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016)*, pages 193–201, 2016.

[18] M. C. Cooper, A. Herzig, F. Maffre, F. Maris, and P. Régnier. Simple epistemic planning: generalised gossiping. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016)*, pages 1563–1564, 2016.

[19] H. v. Ditmarsch, D. Grossi, A. Herzig, W. v. d. Hoek, and L. B. Kuijer. Parameters for epistemic gossip problems. In *Proc. LOFT 2016*, 2016.

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

[21] O. Gasquet, V. Goranko, and F. Schwarzentruber. Big brother logic: logical modeling and reasoning about agents equipped with surveillance cameras in the plane. In A. L. C. Bazzan, M. N. Huhns, A. Lomuscio, and P. Scerri, editors, *Proceedings of the 10th International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14*, pages 325–332. IFAAMAS/ACM, 2014.

[22] J. Gerbrandy. *Bisimulations on Planet Kripke*. PhD thesis, University of Amsterdam, 1999.

[23] J. Gerbrandy and W. Groeneveld. Reasoning about information change. *J. of Logic, Language and Information*, 6(2), 1997.

[24] S. Ghosh, T. Halder, K. Sharma, and R. Verbrugge. Human strategic reasoning in dynamic games: Experiments, logics, cognitive models. In van der Hoek et al. [34], pages 116–128.

[25] A. Hajnal, E. C. B. Milner, and E. Szemerédi. A cure for the telephone disease. *Canadian Mathematical Bulletin*, 15(3):447–450, 1972.

[26] A. Herzig, E. Lorini, and F. Maffre. A poor man's epistemic logic based on propositional assignment and higher-order observation. In *International Conference on Logic, Rationality and Interaction (LORI), Taipei, October 28-31, 2015*, number 9394 in LNCS, pages 156–168. Springer Verlag, 2015.

[27] A. Herzig, E. Lorini, F. Maffre, and F. Schwarzentruber. Epistemic boolean games based on a logic of visibility and control. In Kambhampati [29], pages 1116–1122.

[28] C. A. J. Hurkens. Spreading gossip efficiently. *Nieuw Archief voor Wiskunde*, 5/1(2):208–210, 2000.

[29] S. Kambhampati, editor. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*. IJCAI/AAAI Press, 2016.

[30] F. Maffre. *Ignorance is bliss: observability-based dynamic epistemic logics and their applications*. PhD thesis, Ecole doctorale Math., Info. et Télécom. de Toulouse, Sept. 2016.

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

[32] R. Tijdeman. On a telephone problem. *Nieuw Archief voor Wiskunde*, 19(3):188–192, 1971.

[33] J. van Benthem, J. van Eijck, M. Gattinger, and K. Su. Symbolic model checking for dynamic epistemic logic. In van der Hoek et al. [34], pages 366–378.

[34] W. van der Hoek, W. H. Holliday, and W. Wang, editors. *Logic, Rationality, and Interaction - 5th International Workshop, LORI 2015 Taipei, Taiwan, October 28-31, 2015, Proceedings*, volume 9394 of *Lecture Notes in Computer Science*. Springer, 2015.

[35] W. van der Hoek, P. Iliev, and M. Wooldridge. A logic of revelation and concealment. In W. van der Hoek, L. Padgham, V. Conitzer, and M. Winikoff, editors, *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 1115–1122. IFAAMAS, 2012.

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

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

[38] H. van Ditmarsch, J. van Eijck, P. Pardo, R. Ramezanian, and F. Schwarzentruber. Dynamic gossip. *CoRR*, abs/1511.00867, 2015.