How to share knowledge by gossiping

Andreas Herzig and Faustine Maffre

University of Toulouse, IRIT
http://www.irit.fr/LILaC

Abstract. Given n agents each of which has a secret (a fact not known to anybody else), the classical version of the gossip problem is to achieve shared knowledge of all secrets in a minimal number of phone calls. There exist protocols achieving shared knowledge in $2(n-2)$ calls: when the protocol terminates everybody knows all the secrets. 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 requires not only the communication of secrets, but also the communication of knowledge about secrets. We give a protocol that works in $(k+1)(n-2)$ steps and prove that it is correct: it achieves shared knowledge of level k. The proof is presented in a dynamic epistemic logic that is based on the observability of propositional variables by agents.

Keywords: gossip problem, shared knowledge, common knowledge, dynamic epistemic logic

1 Introduction: the gossip problem and its generalization

The original version of the gossip problem goes as follows [1,9].

> 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 from six to arbitrary numbers of agents n. In the literature one can find various protocols achieving the goal in $2(n-2)$ calls. It has been proved that they are optimal: no protocol exists achieving the goal with less calls [3,10,5].

There are contexts where the agents have to achieve higher-order knowledge, typically in order to coordinate some joint action. While after $2(n-2)$ calls all secrets are shared knowledge, they fail to be common knowledge. Unless everybody knows the protocol and there is a global clock, such common knowledge cannot be attained. More modestly, the agents may want to achieve second-order shared knowledge: they may have the goal that everybody *knows* that everybody

knows all secrets. This paper investigates how such higher-order knowledge can be achieved.

Let Agt be the set of all agents. Let us denote the secret of agent i by s_i. To simplify things we suppose that s_i is a proposition that is true. Let us 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

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

and the formula

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

expresses the goal that every agent knows every secret. Let us abbreviate the conjunction $\bigwedge_{j\in Agt} s_j$ of all secrets by All. Furthermore, let $EK_J\varphi$ abbreviate the conjunction $\bigwedge_{i\in J} K_i\varphi$, where $J \subseteq Agt$ is an arbitrary nonempty subset of Agt. So $EK_{Agt}All$ expresses that all secrets are shared knowledge: every agent knows every secret. $EK_{Agt}EK_{Agt}All$ expresses the goal that every agent knows that all secrets are shared knowledge. The formula

$$\underbrace{EK_{Agt}\ldots EK_{Agt}}_{k \text{ times}} All$$

expresses that all secrets are shared knowledge up to depth $k \geq 1$.

The result of a phone call between two agents is that their knowledge increases. Let us model this by means of modal operators of action: the formula $[Call^i_j]\varphi$ expresses that φ is true after i and j talked to each other. Then $[Call^i_j]EK_{\{i,j\}}(s_i \wedge s_j)$ expresses that the result of $Call^i_j$ is that i and j know their secrets. When we say that during a call the agents communicate all they know then this not only concerns secrets, but also knowledge about secrets and more generally higher-order knowledge. Therefore calls achieve common knowledge between the calling agents, i.e.,

$$[Call^i_j]EK_{\{i,j\}}\ldots EK_{\{i,j\}}(s_i \wedge s_j)$$

is the case for arbitrary nestings of $EK_{\{i,j\}}$. Furthermore, the formula

$$\underbrace{[Call^{i_1}_{j_1}]\ldots[Call^{i_{2(n-2)}}_{j_{2(n-2)}}]}_{2(n-2) \text{ times}} EK_{Agt}All$$

expresses that the protocol where i_1 calls j_1 first, then i_2 calls j_2, ..., and finally $i_{2(n-2)}$ calls $j_{2(n-2)}$ achieves shared knowledge.

We note (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 $(1, 6)$. We are going to introduce a protocol achieving shared knowledge of depth k in $(k{+}1)(n{-}2)$ calls. 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 $i \in Agt$, and dynamic operators $[Call^i_j]$, for $i, j \in Agt$. We had introduced and studied that logic in [6], building on previous work by van der Hoek and colleagues [8,7]. We do not address the question whether our protocol is optimal and leave that to future work.

The paper is organized as follows. Section 2 presents our algorithm. Section 3 recalls syntax and semantics of our dynamic epistemic logic DEL-PAO. In Section 4 we show how to capture the algorithm as a DEL-PAO program. In Section 5 we prove in DEL-PAO that the algorithm is correct. Section 6 concludes.

2 An algorithm achieving higher-order shared knowledge

The following algorithm generates a sequence of calls for a given instance (k, n) of the generalized gossip problem, for $k \geq 1$ and $n \geq 4$. Throughout the algorithm two of the agents, which we call *left* and *right*, will have 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 *left* and *right* talked to play a crucial role. These two further semi-experts 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 $n{-}3{-}t \pmod{n{-}2}$.

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

Figure 1 gives a visual representation of Algorithm 1: agents 0, 1, ..., $n{-}3$ are put on a wheel which, between each turn, rotates clockwise. Agent *left* calls everyone in ascending order, except the agent at the rightmost position of the wheel, then *right* calls this agent.

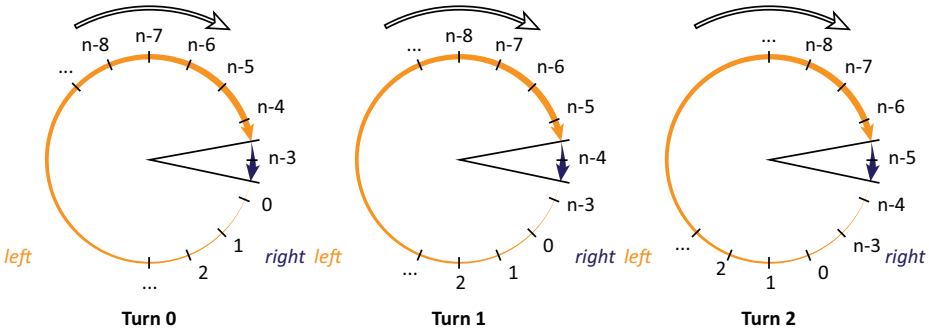

Fig. 1. Graphical represention of the first three turns of Algorithm 1.

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

Theorem 1. *The instance* (k, n) *of the generalized gossip problem can be solved in at most* $(k+1)(n-2)$ *calls.*

The rest of the paper is devoted to the proof of the above theorem: we are going to establish that the sequence of calls produced by the algorithm is indeed a solution. Our proof will be done in the formal language of DEL-PAO that we introduce first.

3 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) [8] and Logic of Revelation and Concealment LRC [7]. Basically 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. In this section, we recall this logic; more details can be found in [6].

3.1 Observability atoms

The atomic formulas of DEL-PAO are called *visibility atoms* and take the form $\mathsf{S}_{i_1}\mathsf{S}_{i_2}...\mathsf{S}_{i_n}p$, where p is a propositional variable from a countable non-empty set *Prop* and $i_1, i_2, ..., i_n$ are agents from a finite non-empty set *Agt*. When n=0 then we have nothing but a propositional variable. For n=1, the atom $\mathsf{S}_{i_1}p$

reads "agent i_1 sees the value of the variable p", and for n=2, the second-order observation $\mathtt{S}_{i_1}\,\mathtt{S}_{i_2}\,p$ 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 $\mathtt{JS}p$ reads "all agents jointly see the value of p". Metaphorically, joint attention about a propositional variable p is the case when there is eye contact between the agents when observing p. Joint visibility implies individual visibility: when a valuation contains $\mathtt{JS}\,p$ then it should also contain $\mathtt{S}_i\,p$.

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:

$$\begin{aligned} K_i p &\leftrightarrow p \wedge \mathtt{S}_i\,p \\ K_i \neg p &\leftrightarrow \neg p \wedge \mathtt{S}_i\,p \\ \neg K_i p \wedge \neg K_i \neg p &\leftrightarrow \neg \mathtt{S}_i\,p \\ K_j K_i p &\leftrightarrow p \wedge \mathtt{S}_i\,p \wedge \mathtt{S}_j\,p \wedge \mathtt{S}_j\,\mathtt{S}_i\,p \\ K_j K_i \neg p &\leftrightarrow \neg p \wedge \mathtt{S}_i\,p \wedge \mathtt{S}_j\,p \wedge \mathtt{S}_j\,\mathtt{S}_i\,p \end{aligned}$$

Formally the definition of observability atoms is as follows. First, the set of *observability operators* is

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

where $\mathtt{S}_i$ stands for individual visibility of agent i and $\mathtt{JS}$ stands for joint visibility of all agents. The set of all sequences of visibility operators is noted OBS^* and the set of all non-empty sequences is noted OBS^+. We use $\sigma\,, \sigma\,', \ldots$ for elements of OBS^*. Finally, 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 $\alpha, \alpha', \ldots, \beta, \beta', \ldots$.

3.2 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:

$$\begin{aligned} \varphi &::= \alpha \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid CK\varphi \mid [\pi]\varphi \\ \pi &::= +\alpha \mid -\alpha \mid \pi;\pi \mid \pi \sqcup \pi \mid \varphi? \end{aligned}$$

where α ranges over ATM and i over Agt.

Our atomic programs are assignments of truth values to atoms from ATM: $+\alpha$ makes α true and $-\alpha$ makes α false. Complex programs are constructed with dynamic logic operators: $\pi;\pi'$ is sequential composition, $\pi \sqcup \pi'$ is nondeterministic

choice, and $\varphi?$ is test. Just as in dynamic logic, the formula $[\pi]\varphi$ reads "after every execution of π, φ is true". The formula $K_i\varphi$ reads "i knows that φ is true on the basis of what she observes", and $CK\varphi$ reads "all agents jointly know that φ is true on the basis of what they jointly observe". 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 [4].

The other boolean operators $\top$, $\bot$, $\vee$, $\rightarrow$ and $\leftrightarrow$ are defined as abbreviations, and $\widehat{K}_i\varphi$ abbreviates $\neg K_i \neg\varphi$. For $J \subseteq Agt$, the shared knowledge modality is defined by

$$EK_J\varphi \stackrel{\text{def}}{=} \bigwedge_{i\in J} K_i\varphi$$

and the iteration of that operator is defined inductively for $k \geq 0$ by $EK_J^0\varphi = \varphi$ and $EK_J^{n+1}\varphi = EK_J EK_J^n\varphi$. Moreover, *skip* abbreviates $\top?$ and *fail* abbreviates $\bot?$. We also use the abbreviation π^k, for $k \geq 0$, inductively defined by $\pi^0 = \mathit{skip}$ and $\pi^{k+1} = \pi^k;\pi$. We sometimes drop set parentheses and, e.g., write $EK_{i,j}\varphi$ instead of $EK_{\{i,j\}}\varphi$.

3.3 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 $i \in Agt$:

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

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

(C1) is about introspection of individual sight: an agent always sees whether she sees the value of an atom. (C2) requires the same for joint sight; indeed, if $\mathtt{JS}\,\alpha$ 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. Together with (C2), (C5) guarantees that $\mathtt{JS}\,\alpha \in V$ implies $\mathtt{JS}\,\sigma\,\alpha \in V$ for $\sigma \in OBS^*$.

The constraints (C4) and (C5) ensure 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 $\sigma \in OBS^+$. Closure under introspective consequence characterizes introspective valuations.

Proposition 1 ([6]). *A valuation $V \subseteq ATM$ 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 $\sigma \in OBS^+$.

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 [6] that the binary relations $\sim_i$ and $\sim_{Agt}$ are equivalence relations on the set of introspective valuations *INTR* and that valuations in *INTR* are not related to valuations outside of *INTR* by $\sim_i$ and $\sim_{Agt}$.

Truth conditions and validity. Given an introspective valuation V, update operations add or remove atoms from V. This requires some care because the resulting valuation should be 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:

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

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

Now the truth conditions are as follows:

$$V \models \alpha \qquad \text{iff } \alpha \in V$$

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

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

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

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

Like $\sim_i$ and $\sim_{Agt}$, it is proven in [6] that valuations in *INTR* are only related to valuations in *INTR* by R_π. Therefore there is no risk to "go out" of the set of introspective valuations with modal operators.

A *model* of φ is a valuation V such that $V \models \varphi$. A formula φ is *satisfiable in INTR* if φ has an introspective model. For example, $\mathtt{JS}\,p \wedge \neg\mathtt{S}_i\,p$ has a model, but does not have an introspective model and is therefore unsatisfiable in *INTR*. A formula φ is *valid in INTR* if if every introspective valuation is a model of φ. We also say that φ is a *validity of* DEL-PAO. For example, $\neg[-\mathtt{S}_1\,\mathtt{S}_2\,p]\mathtt{JS}\,p$ is valid in *INTR*, and $\neg\beta \to [+\alpha]\neg\beta$ is valid in *INTR* if and only if $\alpha \not\rightsquigarrow \beta$.

4 Expressing calls in the language of DEL-PAO

The logic DEL-PAO provides a suitable framework to model calls between agents and to reason about the evolution of their knowledge. Before the proof of correctness of our algorithm, we show how to express calls and we give some of their properties.

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

Formally, let the level k of intended shared knowledge be given. Let i and j be two agents. For a given integer m, let the set all nonempty sequences of

visibility operators $\mathtt{S}_i$ and $\mathtt{S}_j$ of length at most $k-m$ be $\{\sigma_1, \ldots, \sigma_l\}$. For example, for $k = 3$ and $m = 1$ that set is $\{\mathtt{S}_i, \mathtt{S}_j, \mathtt{S}_i\,\mathtt{S}_i, \mathtt{S}_i\,\mathtt{S}_j, \mathtt{S}_j\,\mathtt{S}_i, \mathtt{S}_j\,\mathtt{S}_j\}$. Then $Call^i_j$ is the sequential composition of programs of the form

$$\big(K_iK_{y_1}K_{y_2}\ldots K_{y_m}s \vee K_jK_{y_1}K_{y_2}\ldots K_{y_m}s?; {+}\sigma_1\mathtt{S}_{y_1}\ \ldots\ \mathtt{S}_{y_m}\,s; \ldots; {+}\sigma_l\mathtt{S}_{y_1}\ \ldots\ \mathtt{S}_{y_m}\,s\big)$$
$$\sqcup\ \neg(K_iK_{y_1}K_{y_2}\ldots K_{y_m}s \vee K_jK_{y_1}K_{y_2}\ldots K_{y_m}s)?$$

for secret s in $\{s_i : i \in Agt\}$, integer $m \leq k-1$, and agents $\langle y_1, \ldots, y_m\rangle \in Agt^m$. For example, for $k = 3$ the following is an element of the sequence:

$$\big(K_iK_ys \vee K_jK_ys?; {+}\mathtt{S}_i\,\mathtt{S}_y\,s; {+}\mathtt{S}_j\,\mathtt{S}_y\,s; {+}\mathtt{S}_i\,\mathtt{S}_i\,\mathtt{S}_y\,s; {+}\mathtt{S}_i\,\mathtt{S}_j\,\mathtt{S}_y\,s; {+}\mathtt{S}_j\,\mathtt{S}_i\,\mathtt{S}_y\,s; {+}\mathtt{S}_j\,\mathtt{S}_j\,\mathtt{S}_y\,s\big)$$
$$\sqcup\ \neg(K_iK_ys \vee K_jK_ys)?$$

That piece of program tests whether K_ys is known by i or j and if so makes $\mathtt{S}_y\,s$ visible for both i and j and i's observation of $\mathtt{S}_y\,s$ visible for j, and vice versa; when neither i nor j knows K_ys then the first test $K_iK_ys \vee K_jK_ys?$ fails and the second test $\neg(K_iK_ys \vee K_jK_ys)?$ succeeds and the program does nothing. We 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 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 DEL-PAO 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 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 $K_{i_1}\ldots K_{i_m}s$ or $EK_{J_1}\ldots EK_{J_m}s$. Then:*

1. *$\varphi \to [Call^i_j]\varphi$ is DEL-PAO valid;*
2. *$K_i\varphi \to [Call^i_j]EK^{k-m}_{\{i,j\}}\varphi$ is DEL-PAO valid.*

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

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

5 Correctness of the algorithm

We now prove that the algorithm returns a solution. 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.

Let $Agt = \{left, right, 0, \ldots, n{-}3\}$ be the set of agents and $Prop = \{s_i : i \in Agt\}$ the set of propositional variables. The initial state is modeled by the valuation

$$w_0 = \{s_i : i \in Agt\} \cup \{\mathtt{S}_i\, s_i : i \in Agt\} \cup \{\alpha : \alpha \text{ is valid in } INTR\}.$$

So all secrets are true, each agent knows its own secret, and moreover the introspectively valid atoms are true. We have:

$$w_0 \models \bigwedge_{i \in Agt} K_i\Big(s_i \wedge \bigwedge_{j \in Agt, j \neq i} \neg K_j s_i\Big).$$

An agent is an *expert for depth* t if its personal goal for depth t is reached. Precisely, at w agent i is an expert for depth $t \geq 1$ if and only if

$$w \models K_i E K_{Agt}^{t-1} All.$$

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:

$$\mathsf{compl}_t(i,j) \stackrel{\text{def}}{=} [Call_j^i] E K_{i,j} E K_{Agt}^{t-1} All.$$

Furthermore, two pairs of agents (i_1, i_2) and (j_1, j_2) are complementary for depth t at valuation w if and only if

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

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 1. *We have:*

$$w_0 \models [\mathsf{turn}_0]\big(E K_{left, n-4}(s_{left} \wedge s_0 \wedge \ldots \wedge s_{n-4}) \wedge E K_{right, n-3}(s_{right} \wedge s_{n-3})\big).$$

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

$$left0, left1, \ldots, left(n{-}4), right(n{-}3).$$

By Proposition 2.2 we have $w_0 \models [Call_0^{left}] E K_{left,0}(s_{left} \wedge s_0)$ and therefore $w_0 \models [Call_0^{left}] K_{left}(s_{left} \wedge s_0)$. We do the same for the next call:

$$\begin{aligned} & w_0 \models [Call_0^{left}][Call_1^{left}] E K_{left,1}(s_{left} \wedge s_0 \wedge s_1) \\ \Rightarrow\ & w_0 \models [Call_0^{left}][Call_1^{left}] K_{left}(s_{left} \wedge s_0 \wedge s_1). \end{aligned}$$

And so on until:

$$w_0 \models [Call_0^{left}][Call_1^{left}] \ldots [Call_{n-4}^{left}] E K_{left, n-4}(s_{left} \wedge s_0 \wedge s_1 \wedge \ldots \wedge s_{n-4}).$$

In the same vein we also have $w_0 \models [Call^{right}_{n-3}] EK_{right,n-3}(s_{right} \wedge s_{n-3})$.

By Proposition 2.1 we then obtain:

$$w_0 \models [Call^{left}_0] \ldots [Call^{left}_{n-4}][Call^{right}_{n-3}]\big(EK_{left,n-4}(s_{left} \wedge s_0 \wedge \ldots \wedge s_{n-4}) \wedge EK_{right,n-3}(s_{right} \wedge s_{n-3})\big)$$
$$\Leftrightarrow \; w_0 \models [\mathsf{turn}_0]\big(EK_{left,n-4}(s_{left} \wedge s_0 \wedge \ldots \wedge s_{n-4}) \wedge EK_{right,n-3}(s_{right} \wedge s_{n-3})\big).$$

□

We now characterize the turns after turn_0.

Lemma 2. *For $t \geq 1$, we have:*

$$w_0 \models [\mathsf{turn}_0; \ldots; \mathsf{turn}_t]\big(EK_{left,n-4-t} EK_{left,0-t,\ldots,n-4-t} EK^{t-1}_{Agt} All \wedge$$
$$EK_{right,n-3-t} EK_{right,n-3-t} EK^{t-1}_{Agt} All\big).$$

Proof. We use by induction on t. Both cases resemble the proof of Lemma 1.

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

$$left(n{-}3), left0, left1, \ldots, left(n{-}5), right(n{-}4).$$

By Lemma 1 and Proposition 2.2 we have:

$$w_0 \models [\mathsf{turn}_0][Call^{left}_{n-3}] EK_{left,n-3} EK_{left,n-3} All,$$

from which follows:

$$w_0 \models [\mathsf{turn}_0][Call^{left}_{n-3}] K_{left} EK_{left,n-3} All.$$

Then again by Proposition 2.2:

$$w_0 \models [\mathsf{turn}_0][Call^{left}_{n-3}][Call^{left}_0] EK_{left,0} EK_{left,n-3,0} All$$
$$\Rightarrow w_0 \models [\mathsf{turn}_0][Call^{left}_{n-3}][Call^{left}_0] K_{left} EK_{left,n-3,0} All,$$

and for the next call:

$$w_0 \models [\mathsf{turn}_0][Call^{left}_{n-3}][Call^{left}_0][Call^{left}_1] EK_{left,1} EK_{left,n-3,0,1} All$$
$$\Rightarrow w_0 \models [\mathsf{turn}_0][Call^{left}_{n-3}][Call^{left}_0][Call^{left}_1] K_{left} EK_{left,n-3,0,1} All,$$

and so on until:

$$w_0 \models [\mathsf{turn}_0][Call^{left}_{n-3}][Call^{left}_0][Call^{left}_1] \ldots [Call^{left}_{n-5}] EK_{left,n-5} EK_{left,n-3,0,1,\ldots,n-5} All.$$

Similarly we have:

$$w_0 \models [\mathsf{turn}_0][Call^{right}_{n-4}] EK_{right,n-4} EK_{right,n-4} All.$$

Finally we obtain the result by Proposition 2.1:

$$w_0 \models [\mathsf{turn}_0][Call^{left}_{n-3}][Call^{left}_0] \ldots [Call^{left}_{n-5}][Call^{right}_{n-4}]\big(EK_{left,n-5} EK_{left,n-3,0,1,\ldots,n-5} All$$
$$\wedge \; EK_{right,n-4} EK_{right,n-4} All\big)$$
$$\Leftrightarrow \; w_0 \models [\mathsf{turn}_0][\mathsf{turn}_1]\big(EK_{left,n-5} EK_{left,n-3,0,1,\ldots,n-5} All$$
$$\wedge \; EK_{right,n-4} EK_{right,n-4} All\big).$$

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 Proposition 2.2 we have:

$$w_0 \models [\mathsf{turn}_0; \ldots; \mathsf{turn}_t][Call^{left}_{n-3-t}] EK_{left,n-3-t} EK_{left,n-3-t} EK_{Agt} EK^{t-1}_{Agt} All,$$

that is:

$$w_0 \models [\mathsf{turn}_0; \ldots; \mathsf{turn}_t][Call^{left}_{n-3-t}] EK_{left,n-3-t} EK_{left,n-3-t} EK^{t}_{Agt} All,$$

which implies:

$$w_0 \models [\mathsf{turn}_0; \ldots; \mathsf{turn}_t][Call^{left}_{n-3-t}] K_{left} EK_{left,n-3-t} EK^{t}_{Agt} All.$$

Then by Proposition 2.1:

$$\begin{aligned} & w_0 \models [\mathsf{turn}_0; \ldots; \mathsf{turn}_t][Call^{left}_{n-3-t}][Call^{left}_{0-t}] EK_{left,0-t} EK_{left,n-3-t,0-t} EK^{t}_{Agt} All \\ \Rightarrow\ & w_0 \models [\mathsf{turn}_0; \ldots; \mathsf{turn}_t][Call^{left}_{n-3-t}][Call^{left}_{0-t}] K_{left} EK_{left,n-3-t,0-t} EK^{t}_{Agt} All, \end{aligned}$$

... and so on until:

$$\begin{aligned} w_0 \models [\mathsf{turn}_0; \ldots; \mathsf{turn}_t][Call^{left}_{n-3-t}][Call^{left}_{0-t}] \ldots [Call^{left}_{n-5-t}] \\ EK_{left,n-5-t} EK_{left,n-3-t,0-t,\ldots,n-5-t} EK^{t}_{Agt} All. \end{aligned}$$

Moreover, by Proposition 2.2:

$$w_0 \models [\mathsf{turn}_0; \ldots; \mathsf{turn}_t][Call^{right}_{n-4-t}] EK_{right,n-4-t} EK_{right,n-4-t} EK_{Agt} EK^{t-1}_{Agt} All,$$

that is:

$$w_0 \models [\mathsf{turn}_0; \ldots; \mathsf{turn}_t][Call^{right}_{n-4-t}] EK_{right,n-4-t} EK_{right,n-4-t} EK^{t}_{Agt} All.$$

We end as usual with Proposition 2.1:

$$\begin{aligned} & w_0 \models [\mathsf{turn}_0; \ldots; \mathsf{turn}_t][Call^{left}_{n-3-t}] \ldots [Call^{left}_{n-5-t}][Call^{right}_{n-4-t}] \\ & \qquad \big(EK_{left,n-5-t} EK_{left,n-3-t,\ldots,n-5-t} EK^{t}_{Agt} All \wedge \\ & \qquad\ EK_{right,n-4-t} EK_{right,n-4-t} EK^{t}_{Agt} All\big) \\ \Leftrightarrow\ & w_0 \models [\mathsf{turn}_0; \ldots; \mathsf{turn}_t][\mathsf{turn}_{t+1}] \\ & \qquad \big(EK_{left,n-5-t} EK_{left,n-3-t,\ldots,n-5-t} EK^{t}_{Agt} All \wedge \\ & \qquad\ EK_{right,n-4-t} EK_{right,n-4-t} EK^{t}_{Agt} All\big), \end{aligned}$$

which is our result for $t+1$. □

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

Proof. From Lemma 2 we can deduce:

$$\begin{aligned} w_0 \models [\mathsf{turn}_0, \ldots, \mathsf{turn}_{t-1}]\big(& K_{left} EK_{left,1-t,\ldots,n-3-t} EK^{t-2}_{Agt} All \wedge \\ & K_{right} EK_{right,0-t} EK^{t-2}_{Agt} All\big). \end{aligned}$$

Applying Proposition 2.2 we obtain:

$$w_0 \models [\mathsf{turn}_0, \ldots, \mathsf{turn}_{t-1}][Call^{left}_{right}] EK_{left,right} EK_{Agt} EK^{t-2}_{Agt} All,$$

that is:

$$w_0 \models [\mathsf{turn}_0, \ldots, \mathsf{turn}_{t-1}][Call^{left}_{right}] EK_{left,right} EK^{t-1}_{Agt} All,$$

which is equivalent to:

$$w_0 \models [\mathsf{turn}_0, \ldots, \mathsf{turn}_{t-1}]\mathsf{compl}_t(left, right).$$

Following 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 4. *The goal for depth t, $EK^t_{Agt}All$, is reached after the turn t of Algorithm 1.*

Proof. The 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 3, after the 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 $EK_{\mathit{left},0-t}EK^{t-1}_{Agt}All$.)

Then, after the $n{-}4$ calls $\mathit{left}(1{-}t), \ldots, \mathit{left}(n{-}4{-}t)$ we get by Proposition 2.2:

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

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 3, agents *right* and $n{-}3{-}t$ become experts for depth t. (Thus $EK_{\mathit{right},n-3-t}EK^{t-1}_{Agt}All$.)

Therefore after the $n{-}2$ calls of the turn t we have $EK_{Agt}EK^{t-1}_{Agt}All$, which is equivalent to $EK^t_{Agt}All$. □

Proposition 3. *The sequence resulting from Algorithm 1 gives a solution to the generalized gossip problem.*

Proof. By Lemma 4, 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. □

6 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.

The gossip problem recently attracted quite some attention in the dynamic epistemic logic community [2]. 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 multiagent planning problems that can be compared to the blocksworld in classical planning.

Acknowledgements

We would like to acknowledge several discussions about the gossip problem at the inspiring August 2015 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 Davide Grossi, Emiliano Lorini and Martin Cooper.

References

1. Akkoyunlu, E.A., Ekanadham, K., Hubert, R.V.: Some constraints and tradeoffs in the design of network communications. In: Proceedings of the 5th ACM Symposium on Operating Systems Principles. pp. 67–74. ACM Press (1975)
2. Attamah, M., van Ditmarsch, H., Grossi, D., van der Hoek, W.: Knowledge and gossip. Proceedings of 21st ECAI pp. 21–26 (2014)
3. Baker, B., Shostak, R.: Gossips and telephones. Discrete Mathematics 2(3), 191–193 (1972)
4. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning about Knowledge. MIT Press (1995)
5. Hajnal, A., Milner, E.C.B., Szemerédi, E.: A cure for the telephone disease. Canadian Mathematical Bulletin 15(3), 447–450 (1972)
6. Herzig, A., Lorini, E., Maffre, F.: A poor man's epistemic logic based on propositional assignment and higher-order observation (regular paper). In: International Conference on Logic, Rationality and Interaction (LORI), Taipei, October 28-31, 2015. Springer Verlag (2015), `http://www.irit.fr/~Andreas.Herzig/P/Lori15.html`
7. van der Hoek, W., Iliev, P., Wooldridge, M.: A logic of revelation and concealment. In: van der Hoek, W., Padgham, L., Conitzer, V., Winikoff, M. (eds.) Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems. pp. 1115–1122. IFAAMAS (2012)
8. van der Hoek, W., Troquard, N., Wooldridge, M.: Knowledge and control. In: Sonenberg, L., Stone, P., Tumer, K., Yolum, P. (eds.) Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems. pp. 719–726. IFAAMAS (2011)
9. Hurkens, C.A.J.: Spreading gossip efficiently. Nieuw Archief voor Wiskunde 5/1(2), 208–210 (2000)
10. Tijdeman, R.: On a telephone problem. Nieuw Archief voor Wiskunde 19(3), 188–192 (1971)