Verification of Distributed Epistemic Gossip Protocols

Krzysztof R. Apt APT@CWI.NL
CWI, Amsterdam, The Netherlands
MIMUW, University of Warsaw, Poland

Dominik Wojtczak D.WOJTCZAK@LIVERPOOL.AC.UK
University of Liverpool, Liverpool, U.K.

Abstract

Gossip protocols aim at arriving, by means of point-to-point or group communications, at a situation in which all the agents know each other secrets. Distributed epistemic gossip protocols use as guards formulas from a simple epistemic logic and as statements calls between the agents. They are natural examples of ***knowledge based programs***.

We prove here that these protocols are implementable, that their partial correctness is decidable and that termination and two forms of fair termination of these protocols are decidable, as well. To establish these results we show that the definition of semantics and of truth of the underlying logic are decidable.

1. Introduction

This paper is concerned with the verification of a specific type of gossip protocols. In such protocols each agent holds a secret initially known only to him. The secrets spread by means of communications. During them, e.g., point-to-point or group communications, the participating agents exchange some, possibly all, secrets they know. The aim of a gossip protocol is to arrive at a situation in which all the agents know each other secrets.

Gossip protocols have been successful in various domains, e.g., communication networks (Hedetniemi, Hedetniemi, & Liestman, 1988), computation of aggregate information (Kempe, Dobra, & Gehrke, 2003), and data replication (Ladin, Liskov, Shrira, & Ghemawat, 1992). A more recent account is given by Hromkovic, Klasing, Pelc, Ruzicka, and Unger (2005) and Kermarrec and van Steen (2007). In these references gossip protocols are viewed as parallel, probabilistic or distributed programs.

As a simple example assume that the set of agents is $\{a, b, c, d, e_1, \ldots, e_{n-4}\}$, where $n \geq 4$, (if $n = 4$ then there are no e_i agents) and assume that the agents communicate by means of phone calls during which they exchange all secrets they know. Then take the call sequence

$$(a, e_1), (a, e_2), \ldots, (a, e_{n-4}).$$

After it agent a knows all the secrets of the agents $e_1, \ldots, e_{n-4}$. We follow it by the call sequence $(a, b), (c, d), (a, d), (b, c)$. After it agents a, b, c, d know all the secrets. So following it by the above call sequence $(a, e_1), (a, e_2), \ldots, (a, e_{n-4})$ we achieve the desired situation in which all agents know all the secrets. This took $2n - 4$ calls.

One of the early results established by a number of authors in the seventies, for instance by Tijdeman (1971), is that for $n \geq 4$ agents at least $2n - 4$ phone calls are needed to reach the above final situation.

The above protocol is centralized in the sense that it consists of a globally scheduled sequences of calls. Attamah, van Ditmarsch, Grossi, and Van der Hoek (2014b) introduced and studied distributed epistemic gossip protocols. 'Distributed' means that each agent acts autonomously, and 'epistemic' means that the gossip protocols refer to agents' knowledge. These protocols were described as formulas in an epistemic dynamic logic. Consequently they are examples of ***knowledge-based programs*** that were introduced by Fagin, Halpern, Moses, and Vardi (1997). These are programs that use tests for knowledge. Examples are protocols for the sequence transmission problem, such as the alternating bit protocol, studied by Halpern and Zuck (1992).

Apt, Grossi, and Van der Hoek (2016) built upon the work of Attamah et al. (2014b) and studied distributed epistemic gossip protocols (from now on just: gossip protocols) for calls of a different type than the ones considered by Attamah et al. (2014b), which in our view are closer to the setting of distributed programming. These gossip protocols are strikingly simple in their syntax (though not semantics): they are just parallel compositions of loops in which the agents repeatedly perform a call assuming the corresponding epistemic formula (a guard) evaluates to true. This considerably simplified the task of their verification. The subject of our paper is to analyze semantics and verification of these gossip protocols. We prove the following results.

- These gossip protocols are implementable.

  More precisely, we show that the semantics of the underlying epistemic logic is decidable and consequently it is decidable to determine whether a guard is true after a sequence of calls.

- Partial correctness of these gossip protocols is decidable.

  More precisely, we show that truth in the underlying epistemic logic is decidable. This implies the claim since partial correctness of a gossip protocol can be expressed as a specific formula in this logic.

- Termination of these gossip protocols is decidable.

- Fair termination (in two different senses) of these gossip protocols is decidable.

  Moreover, we show that the concepts of fairness and justice, that differ for arbitrary nondeterministic and distributed programs, coincide for these gossip protocols.

This implies that distributed epistemic gossip protocols are very specific programs that in particular do not have the full power of Turing machines.

The obtained results, while sufficient for an analysis of the considered gossip protocols, raise a number of interesting open problems concerning both the underlying logic and the protocols and to which we return in the conclusions.

1.1 Related Work

Epistemic reasoning about gossip protocols has been recently investigated from several viewpoints. The stage was set up in the already mentioned work of Attamah et al. (2014b). We shall discuss one specific aspect of this paper, namely the type of calls, in the final section

of the paper. Attamah, van Ditmarsch, Grossi, and Van der Hoek (2014a) presented a tool that given a high level description of an epistemic protocol in the setting of Attamah et al. (2014b) generates the characteristics of the protocol. The calls considered there differ from ours, so this approach is not applicable to our setting. Further, van Ditmarsch, van Eijck, Pardo, Ramezanian, and Schwarzentruber (2017b) undertook a study of dynamic distributed gossip protocols in which the calls allow the agents not only to share the secrets but also to transmit the links. The purpose of that paper is to characterize such protocols in terms of the class of graphs for which they terminate. Such protocols then differ from the ones here considered, which are static.

Next, Herzig and Maffre (2015) and Herzig and Maffre (2017) studied gossip protocols that aim at achieving higher-order shared knowledge, for example knowledge of level 2 which stipulates that everybody knows that everybody knows all secrets. In particular, a protocol is presented and proved correct that achieves in $(k + 1)(n - 2)$ steps shared knowledge of level k. These matters are further investigated by Cooper, Herzig, Maffre, Maris, and Regnier (2016b), where optimal protocols for various versions of such a generalized gossip problem are given depending on various parameters, such as type of the underlying graph or the type of communication. Also gossip problems are studied in which some negative goals, such as that certain agents must not know certain secrets, are supposed to be achieved.

Further, Cooper, Herzig, Maffre, Maris, and Régnier (2016a) studied gossip protocols as an instance of multi-agent epistemic planning that is subsequently translated into the classical planning language PDDL. In turn, van Ditmarsch, Grossi, Herzig, van der Hoek, and Kuijer (2016) presented the gossip problems in an epistemic framework that provides several parameters allowing us to capture such aspects as the initial knowledge of the agents, the type of communication used, and the desired type of the protocol (for example, a symmetric one). For some of the combinations of the parameters the minimum number of calls needed to reach the final situation is established. The expected time of termination of several gossip protocols on completely connected networks was studied by van Ditmarsch, Kokkinis, and Stockmarr (2017a).

Fairness is a widely considered concept in nondeterministic and distributed computing, see, e.g., (Francez, 1986). As shown by Apt, Francez, and Katz (1988), in distributed systems it can be defined in a number of ways.

Finally, let us mention that several decidability results reported here were recently established by Apt and Wojtczak (2017a) for the logic containing the common knowledge operator. Further, Apt, Kopczyński, and Wojtczak (2017) investigated the computational aspects of the problems here studied. Building upon the work here reported it was established that the implementability of a distributed epistemic gossip protocol is a $\mathrm{P}_{\|}^{\mathrm{NP}}$-complete problem, while the problems of its partial correctness and termination are in $\mathrm{coNP}^{\mathrm{NP}}$. The computational analysis of two types of fair termination studied here was not considered.

1.2 Plan

The paper is organized as follows. In the next two sections, 2 and 3, we recall the syntax and semantics of the considered epistemic logic, originally introduced by Apt et al. (2016). Then, in Section 4 we recall the distributed epistemic gossip protocols studied in that paper and in

Section 5 we discuss in detail an example of such a protocol. Next, in Section 6 we introduce an alternative, equivalent, semantics, which helps us to prove the desired decidability results.

In Section 7 we prove the decidability of the problem of checking whether a formula is true after a given sequence of calls. This implies that the considered gossip protocols are implementable. In Section 8 we show that the definition of truth in the considered logic is decidable.

In the subsequent two sections, 9 and 10 we apply these results to a study of gossip protocols. We establish there decidability of four properties of these protocols: partial correctness, termination, and fair termination in two different senses. Finally, in Section 11, we discuss some related open problems and recall other types of communications studied by Attamah et al. (2014b).

2. Syntax

Throughout the paper we assume a fixed finite set A of at least three ***agents*** and a fixed set P of all secrets. We assume that each agent holds exactly one ***secret*** and that there exists a bijection between the set of agents and the set of secrets. Further, each agent is uniquely determined by his secret. To indicate this we denote the secret of agent a by A, the secret of agent b by B and so on.

Our aim is to analyze what the agents know after a sequence of calls took place. So first we introduce the calls and then consider an epistemic language allowing us to refer to agents' knowledge.

Each ***call*** concerns two different agents, the *caller* (a below) and the *callee* (b). Apt et al. (2016) distinguished three ***modes of communication*** of a call:

- ***push-pull***, written as ab or (a, b). After this call the caller and the callee learn each others secrets,
- ***push***, written as $a \triangleright b$. After this call the callee learns all the secrets held by the caller,
- ***pull***, written as $a \triangleleft b$. After this call the caller learns all the secrets held by the callee.

So the push-pull mode describes two-way communication, while the push and the pull modes describe one-way communication. Calls are denoted by c, d. Abusing notation we write $a \in \mathsf{c}$ to denote that agent a is one of the two agents involved in the call c (e.g., for $\mathsf{c} := ab$ we have $a \in \mathsf{c}$ and $b \in \mathsf{c}$). Further, we say that a call is a *b-call* if agent b is involved in it.

The mode of communication is concerned only with the agents involved in the call and states nothing about the effect of the call on the knowledge of other agents. Here we follow the approach of Apt et al. (2016) and stipulate that agents not involved in the call are not aware of it. This will be addressed in Definition 1 below. Other options are discussed in the last section.

To discuss knowledge of the agents we consider formulas in a simple epistemic language $\mathcal{L}$ defined by the following grammar:

$$\phi ::= F_a p \mid \neg\phi \mid \phi \wedge \phi \mid K_a \phi,$$

where $p \in \mathsf{P}$ and $a \in \mathsf{A}$. Each secret is viewed a distinct constant. In the example formulas we shall also use the disjunction $\phi \vee \psi$ as an abbreviation for $\neg(\neg\phi \wedge \neg\psi)$, the implication $\phi \rightarrow \psi$ as an abbreviation for $\neg(\phi \wedge \neg\psi)$, and the equivalence $\leftrightarrow$ as a shorthand for the conjunction of two implications.

We read $F_a p$ as 'agent a is familiar with the secret p' and $K_a\phi$ as 'agent a knows that formula ϕ is true'. So $F_a p$ is an atomic formula, while $K_a\phi$ is a compound formula. In fact, all atomic formulas of $\mathcal{L}$ are of the form $F_a p$.

Attamah et al. (2014b) used instead of the atomic formula $F_a p$ the knowledge formula $K_a p \vee K_a \neg p$, which states that agent a knows the truth value of the proposition p. This leads to a different language in which the atomic formulas are secrets that are viewed as propositional variables. The advantage of the approach adopted by Apt et al. (2016) and followed here is that it simplifies semantics and reasoning about it. Also, it allows one to suppress one level of nesting of the modalities.

In what follows we shall distinguish the following sublanguages of $\mathcal{L}$:

- $\mathcal{L}_{pr}$, its propositional part, which consists of the formulas that do not use the K_a modalities,
- $\mathcal{L}_a$, where $a \in \mathsf{A}$ is a fixed agent, which consists of the formulas the only modality of which is K_a, and in which all atomic formulas are of the form $F_a p$,
- $\mathcal{L}_{wn}$, which consists of the formulas without the nested use of the K_a modalities.

In our logic the formulas of the form $K_a\phi$ and $K_a K_a\phi$ are equivalent, so in the last sublanguage one can also allow the formulas of the form $K_a K_a \ldots K_a\phi$. All gossip protocols studied in the work of Apt et al. (2016) use as guards only formulas from $\mathcal{L}_{wn}$, that is in a program for agent a only guards from $\mathcal{L}_a \cap \mathcal{L}_{wn}$ are used.

3. Semantics

We now recall semantics of the epistemic formulas introduced by Apt et al. (2016). It relies on the concept of a gossip situation.

3.1 Gossip Situations and Their Modifications

A ***gossip situation*** is a sequence $\mathsf{s} = (\mathsf{Q}_a)_{a\in\mathsf{A}}$, where $\mathsf{Q}_a \subseteq \mathsf{P}$ for each agent a. Intuitively, Q_a is the set of secrets a is familiar with in the gossip situation s. The ***initial gossip situation*** is the one in which each Q_a equals $\{A\}$ and is denoted by root. It reflects the fact that initially each agent is familiar only with his own secret. We say that an agent a is an ***expert*** in a gossip situation s if he is familiar in s with all the secrets, i.e., if $\mathsf{Q}_a = \mathsf{P}$.

We will use the following concise notation for gossip situations. Sets of secrets will be written down as lists. e.g., the set $\{A, B, C\}$ will be written as ABC. Gossip situations will be written down as lists of lists of secrets separated by dots. E.g., if there are three agents, a, b and c, then $\mathsf{root} = A.B.C$ and the gossip situation $(\{A, B\}, \{A, B\}, \{C\})$ will be written as $AB.AB.C$.

Each call c transforms the current gossip situation s by modifying the sets of secrets the agents involved in the call are familiar with. This modification depends on the mode of communication. Consider a gossip situation $\mathsf{s} := (\mathsf{Q}_d)_{d\in\mathsf{A}}$.

- Suppose $\mathsf{c} = ab$.

  Then $\mathsf{c}(\mathsf{s}) := (Q'_d)_{d\in\mathsf{A}}$, where $\mathsf{Q}'_a = \mathsf{Q}'_b = \mathsf{Q}_a \cup \mathsf{Q}_b$, and for $c \notin \{a, b\}$, $\mathsf{Q}'_c = \mathsf{Q}_c$.

  So the effect of a push-pull call is that the caller and the callee share the secrets they are familiar with.

- Suppose $\mathsf{c} = a \triangleright b$.

  Then $\mathsf{c}(\mathsf{s}) := (Q'_a)_{a\in\mathsf{A}}$, where $\mathsf{Q}'_b = \mathsf{Q}_a \cup \mathsf{Q}_b$, $\mathsf{Q}'_a = \mathsf{Q}_a$, and for $c \notin \{a, b\}$, $\mathsf{Q}'_c = \mathsf{Q}_c$;

  So the effect of a push call is that the callee learns the secrets of the caller.

- Suppose $\mathsf{c} = a \triangleleft b$.

  Then $\mathsf{c}(\mathsf{s}) := (Q'_a)_{a\in\mathsf{A}}$, where $\mathsf{Q}'_a = \mathsf{Q}_a \cup \mathsf{Q}_b$, $\mathsf{Q}'_b = \mathsf{Q}_b$, and for $c \notin \{a, b\}$, $\mathsf{Q}'_c = \mathsf{Q}_c$.

  So the effect of a pull call is that the caller learns the secrets of the callee.

3.2 Call Sequences

Apt et al. (2016) studied computations of the gossip protocols, so both finite and infinite call sequences were used. Here we focus mostly on the finite call sequences. So to be brief, unless explicitly stated, a ***call sequence*** is assumed to be finite. For simplicity we assume that all calls in a call sequence are of the same mode.

The empty sequence is denoted by ϵ. We use $\mathbf{c}$ to denote a call sequence and $\mathbf{C}$ to denote the set of all finite call sequences. Given call sequences $\mathbf{c}$ and $\mathbf{d}$ and a call c we denote by $\mathbf{c}.\mathsf{c}$ the outcome of adding c at the end of the sequence $\mathbf{c}$ and by $\mathbf{c}.\mathbf{d}$ the outcome of appending the sequences $\mathbf{c}$ and $\mathbf{d}$. We write $\mathbf{c} \preceq \mathbf{d}$ to denote the fact that $\mathbf{d}$ extends $\mathbf{c}$, i.e., that for some $\mathbf{c}'$ we have $\mathbf{c}.\mathbf{c}' = \mathbf{d}$. We have thus two equivalent ways of representing call sequences, either as $(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_n)$ or as $\mathsf{c}_1.\mathsf{c}_2.\ldots.\mathsf{c}_n$.

The result of applying a call sequence to a situation s is defined inductively as follows:

$$\epsilon(\mathsf{s}) := \mathsf{s},\ (\mathsf{c}.\mathbf{c})(\mathsf{s}) := \mathbf{c}(\mathsf{c}(\mathsf{s})).$$

Example 1. Let $\mathsf{A} = \{a, b, c\}$. Consider the call sequence (ac, bc, ac). It generates the following successive gossip situations starting from root:

$$A.B.C \xrightarrow{ac} AC.B.AC \xrightarrow{bc} AC.ABC.ABC \xrightarrow{ac} ABC.ABC.ABC.$$

Hence $(ac, bc, ac)(\mathsf{root}) = (ABC.ABC.ABC)$.

Next, consider the call sequence $(a \triangleleft c, b \triangleleft c, a \triangleleft c)$. It generates the following successive gossip situations starting from root:

$$A.B.C \xrightarrow{a\triangleleft c} AC.B.C \xrightarrow{b\triangleleft c} AC.BC.C \xrightarrow{a\triangleleft c} AC.BC.C.$$

Hence $(a \triangleleft c, b \triangleleft c, a \triangleleft c)(\mathsf{root}) = (AC.BC.C)$.

Finally, consider the call sequence $(a \triangleright b, b \triangleright c, c \triangleright a)$. It generates the following successive gossip situations starting from root:

$$A.B.C \xrightarrow{a\triangleright b} A.AB.C \xrightarrow{b\triangleright c} A.AB.ABC \xrightarrow{c\triangleright a} ABC.AB.ABC.$$

Hence $(a \triangleright b, b \triangleright c, c \triangleright a)(\mathsf{root}) = (ABC.AB.ABC)$. □

3.3 Gossip Models and Truth

A gossip situation is a set of possible combinations of secret distributions among the agents. As calls progress in sequence from the initial situation, agents may be uncertain about which call sequence took place. This uncertainty is captured by the appropriate equivalence relations on the call sequences. Suppose first that the mode of communication is push-pull.

Definition 1. *The **gossip model** is a tuple $\mathcal{M} := (\mathbf{C}, \{\sim_a\}_{a\in\mathsf{A}})$, where each $\sim_a \subseteq \mathbf{C}\times\mathbf{C}$ is the smallest relation satisfying the following conditions:*

[Base] $\epsilon \sim_a \epsilon$,

[Step] *Suppose* $\mathbf{c} \sim_a \mathbf{d}$.

(i) If $a \notin \mathsf{c}$, then $\mathbf{c}.\mathsf{c} \sim_a \mathbf{d}$ and $\mathbf{c} \sim_a \mathbf{d}.\mathsf{c}$.

(ii) If there exists $b \in \mathsf{A}$ and $\mathsf{c} \in \{ab, ba\}$ such that $\mathbf{c}.\mathsf{c}(\mathsf{root})_a = \mathbf{d}.\mathsf{c}(\mathsf{root})_a$, then $\mathbf{c}.\mathsf{c} \sim_a \mathbf{d}.\mathsf{c}$.

In *(i)* we formalize the assumption that the agents are not aware of the calls they do not participate in. In turn, in *(ii)* we capture the intuition that two call sequences are indistinguishable for an agent if the sets of his calls in both sequences are the same and in each sequence he observes the same set of secrets.

Note that according to our definition, for $a \neq b$, $ab \not\sim_a ba$. This means that agents are aware of who is calling whom and can differentiate between calls in which these roles are reversed. Apt et al. (2016) used a slightly different definition of *(ii)* according to which $ab \sim_a ba$. We consider the above definition more intuitive.

To illustrate this definition consider the following examples. Based on *(ii)* we have $ab \sim_a ab$, so by *(i)* used twice we have $(ab, bc) \sim_a (ab, bd)$. But we do not have $(ab, bc) \sim_a (ba, bd)$, because $ab \not\sim_a ba$. We also do not have $(bc, ab) \sim_a (bd, ab)$ since $(bc, ab)(\mathsf{root})_a = ABC \neq ABD = (bd, ab)(\mathsf{root})_a$. At the same time, we have by *(i)* used four times $(bc, bd) \sim_a (cd, bc)$, so by *(ii)* $(bc, bd, ab) \sim_a (cd, bc, ab)$.

Later, in Section 6, we introduce an alternative definition of the relations $\sim_a$ that provides additional insights in the above definition and in particular will allow us to prove that each $\sim_a$ is an equivalence relation.

Suppose now that the mode of communication is push. Then Definition 1 needs to be modified as follows. The original clause (ii) is replaced by the following clause:

(ii') If there exists $b \in \mathsf{A}$ and $\mathsf{c} \in \{a \triangleright b, b \triangleright a\}$ such that $\mathbf{c}.\mathsf{c}(\mathsf{root})_a = \mathbf{d}.\mathsf{c}(\mathsf{root})_a$, then $\mathbf{c}.\mathsf{c} \sim_a \mathbf{d}.\mathsf{c}$.

Note that when b and c are different agents, then $a \triangleright b \not\sim_a a \triangleright c$. The intuition is that agent a is fully aware of the calls he performed. The calls $a \triangleright b$ and $a \triangleright c$ are different for him even though he does not learn anything from any of them.

For instance, by (i) and (ii') we have $(a \triangleright b, b \triangleright c) \sim_a (b \triangleright c, a \triangleright b)$. But we do not have $(b \triangleright a, c \triangleright b) \sim_a (c \triangleright b, b \triangleright a)$, since $(b \triangleright a, c \triangleright b)(\mathsf{root})_a = AB \neq ABC = (c \triangleright b, b \triangleright a)(\mathsf{root})_a$.

Finally, suppose that the mode of communication is pull. Then Definition 1 needs to be modified as follows. The original clause (ii) is replaced by the following clause:

(ii'') If there exists $b \in \mathsf{A}$ and $\mathsf{c} \in \{a \lhd b, b \lhd a\}$ such that $\mathbf{c}.\mathsf{c}(\mathsf{root})_a = \mathbf{d}.\mathsf{c}(\mathsf{root})_a$, then $\mathbf{c}.\mathsf{c} \sim_a \mathbf{d}.\mathsf{c}$.

For instance, by (i) and (ii'') we have $(b \lhd a, c \lhd b) \sim_a (c \lhd b, b \lhd a)$. But we do not have $(a \lhd b, b \lhd c) \sim_a (b \lhd c, a \lhd b)$ since $(a \lhd b, b \lhd c)(\mathsf{root})_a = AB \neq ABC = (b \lhd c, a \lhd b)(\mathsf{root})_a$.

Now that we provided the definition of a model we recall the definition of truth, which is the same for all three modes of communication.

Definition 2. *Consider the gossip model $\mathcal{M} := (\mathbf{C}, (\sim_a)_{a \in \mathsf{A}})$ and a call sequence $\mathbf{c} \in \mathbf{C}$. We define the satisfaction relation $\models$ inductively as follows (clauses for Boolean connectives are as usual and omitted):*

$$(\mathcal{M}, \mathbf{c}) \models F_a p \quad \textit{iff} \quad p \in \mathbf{c}(\mathsf{root})_a,$$
$$(\mathcal{M}, \mathbf{c}) \models K_a \phi \quad \textit{iff} \quad \forall \mathbf{d} \textit{ s.t. } \mathbf{c} \sim_a \mathbf{d},\ (\mathcal{M}, \mathbf{d}) \models \phi.$$

Further we define

$$\mathcal{M} \models \phi \quad \textit{iff} \quad \forall \mathbf{c}\ (\mathcal{M}, \mathbf{c}) \models \phi.$$

*When $(\mathcal{M}, \mathbf{c}) \models \phi$ we say that ϕ is **true after** $\mathbf{c}$ and when $\mathcal{M} \models \phi$ we say that ϕ is **true**.*

So a formula $F_a p$ is true after $\mathbf{c}$ whenever secret p belongs to the set of secrets agent a is familiar with in the situation generated by the call sequence $\mathbf{c}$ applied to the initial situation root. The knowledge operator is interpreted as customary in epistemic logic, using the equivalence relations $\sim_a$.

4. Gossip Protocols

Apt et al. (2016), as a follow up on the work of Attamah et al. (2014b), studied distributed epistemic gossip protocols. Their goal is to reach a gossip situation in which each agent is an expert. In other words, their goal is to transform a gossip situation in which the formula $\bigwedge_{a \in \mathsf{A}} F_a A \wedge \bigwedge_{a,b \in \mathsf{A}, a \neq b} \neg F_a B$ is true into one in which the formula $\bigwedge_{a,b \in \mathsf{A}} F_a B$ is true. Let us recall their definition.

By a ***component program***, in short a ***program***, for an agent a we mean a statement of the form

$$*[[]_{j=1}^{m}\ \psi_j \to \mathsf{c}_j],$$

where $m > 0$ and each $\psi_j \to \mathsf{c}_j$ is such that $\psi_j \in \mathcal{L}_a$ and a is the caller in the call c_j.

Given a formula $\psi \in \mathcal{L}_a$ and a call c, we call the construct $\psi \to \mathsf{c}$ a ***rule*** and refer in this context to ψ as a ***guard***. Intuitively, $*$ denotes a repeated execution of the rules, one at a time, where each time non-deterministically a rule is selected whose guard is true. Finally, by a ***distributed epistemic gossip protocol***, in short a ***gossip protocol***, we mean a parallel composition of component programs, one for each agent.

This syntax loosely follows the syntax of the language CSP (Communicating Sequential Processes) of Hoare (1978) that extends the guarded command language of Dijkstra (1975) by disjoint parallel composition and commands for synchronous communication. CSP was realized in the distributed programming language OCCAM (see INMOS in INMOS Limited, 1984).

Gossip protocols here considered can be seen special cases of knowledge-based programs introduced by Fagin et al. (1997). This means that one could instantiate their definition of global states, runs and actions in our setup. However, the framework we consider here is very simple (for example assignments are not allowed and only two agents are involved in each action), so a much simpler, direct, definition of computations can be given.

Assume now a gossip protocol P that is a parallel composition of the component programs $*[[]_{j=1}^{m_a}\ \psi_j^a \to \mathsf{c}_j^a]$, one for each agent $a \in \mathsf{A}$.

The ***computation tree*** of P is a directed tree defined inductively as follows. Its nodes are call sequences and its root is the empty call sequence ϵ. Further, if $\mathbf{c}$ is a node and for some rule $\psi_j^a \to \mathsf{c}_j^a$ we have $(\mathcal{M}, \mathbf{c}) \models \psi_j^a$, then $\mathbf{c}.\mathsf{c}_j^a$ is a node that is a direct descendant of $\mathbf{c}$. Intuitively, the arc from $\mathbf{c}$ to $\mathbf{c}.\mathsf{c}_j^a$ records the effect of the execution of the rule $\psi_j^a \to \mathsf{c}_j^a$ performed after the call sequence $\mathbf{c}$ took place.

By a ***computation*** of a gossip protocol we mean a maximal rooted path in its computation tree. This tree is finitely branching, so by König's Lemma (König, 1927) if it has arbitrary long paths, then it also has infinite paths.

We say that the gossip protocol P is ***partially correct*** if for all leafs $\mathbf{c}$ of the computation tree of P

$$(\mathcal{M}, \mathbf{c}) \models \bigwedge_{a,b \in \mathsf{A}} F_a B, \tag{1}$$

i.e., if each agent is an expert in the gossip situation $\mathbf{c}(\mathsf{root})$.

Note that a rooted finite path χ in the computation tree of P is a finite computation iff its leaf $\mathbf{c}$ cannot be extended by any call, so iff

$$(\mathcal{M}, \mathbf{c}) \models \bigwedge_{a \in \mathsf{A}} \bigwedge_{j=1}^{m_a} \neg \psi_j^a.$$

We call the formula $\bigwedge_{a \in \mathsf{A}} \bigwedge_{j=1}^{m_a} \neg \psi_j^a$ the ***exit condition*** of the gossip protocol P. So P is partially correct iff the implication

$$\bigwedge_{a \in \mathsf{A}} \bigwedge_{j=1}^{m_a} \neg \psi_j^a \to \bigwedge_{a,b \in \mathsf{A}} F_a B \tag{2}$$

is true after every call sequence corresponding to a node of the computation tree for P. In particular if this implication is true, then P is partially correct.

We say furthermore that P ***terminates*** if all its computations are finite. We also consider two variants of termination. To define them we need a subsidiary notion. We call a rule ***enabled*** after a call sequence $\mathbf{c}$ if its guard is true after $\mathbf{c}$. Given a gossip protocol we say that an agent is ***enabled*** after a call sequence $\mathbf{c}$ if one of the rules in its program is enabled.

We now stipulate that each finite computation is ***rule-fair*** and ***agent-fair***. An infinite computation is ***rule-fair*** (resp. ***agent-fair***) if all rules (resp. agents) that are enabled after infinitely many prefixes (in short, infinitely often) are selected infinitely often. We say that a gossip protocol P ***rule-fairly terminates*** (resp. ***agent-fairly terminates***) if all its rule-fair (resp. agent-fair computations) are finite. Agent-fairness was introduced by Apt et al. (2016), where it was simply called fairness.

5. Example: a Protocol Over Undirected Graphs

To illustrate the power of gossip protocols consider the following example taken from the work of Apt et al. (2017). Suppose that the agents are nodes of an undirected connected graph (V, E) and that the calls can take place only between pairs of agents connected by an edge. Let N_a denote the set of neighbours of node a.

Consider a gossip protocol P with the following program for agent a:

$$*[[]_{b \in N_a, C \in \mathsf{P}} F_a C \wedge \neg K_a F_b C \to ab].$$

Informally, agent a calls a neighbour b if a is familiar with some secret (here C) and he does not know whether b is familiar with it.

Despite its simplicity this protocol can exhibit a complex behaviour. Consider for instance the binary tree depicted in Figure 1 connecting agents a, b, c, d, e, f, g.

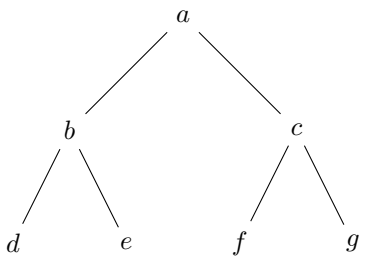

Figure 1: A connection graph

The following two example call sequences can be generated for it by the considered protocol:

- $(ab, ac, bd, ba, ac, be, bd, ba, ac, cf, ca, cg, cf, ca, ab, bd, be)$,
- $(bd, be, ba, ac, cf, cg, cf, ca, ab, bd, be)$.

To prove partial correctness of this protocol consider its exit condition

$$\bigwedge_{(a,b) \in E} \bigwedge_{C \in \mathsf{P}} (F_a C \to K_a F_b C).$$

For all agents a and b and secrets C, the formula $K_a F_b C \to F_b C$ is true, so the exit condition implies

$$\bigwedge_{(a,b) \in E} \bigwedge_{C \in \mathsf{P}} (F_a C \to F_b C).$$

Consider now an agent a and the secret B of agent b. Let $i_1, \ldots, i_h$ be a path that connects b with a. So $i_1 = b$ and $i_h = a$. The above formula implies that for $g \in \{1, \ldots, h-1\}$ we have $\bigwedge_{C \in \mathsf{P}} (F_{i_g} C \to F_{i_{g+1}} C)$. By combining these $h-1$ formulas we get $\bigwedge_{C \in \mathsf{P}} (F_b C \to F_a C)$. But $F_b B$ is true, so we conclude $F_a B$. Consequently $\bigwedge_{a,b \in \mathsf{A}} F_a B$, as desired.

To prove termination it suffices to note that after each call ab the size of the set $\{(a, b, C) \mid \neg K_a F_b C\}$ decreases.

6. An Alternative Equivalence Relation

From now on we focus on the push-pull mode. We provide now an alternative equivalence relation between the call sequences that is easier to work with. Also, it provides another insight into the definition of the $\sim_a$ relations. First we introduce the following notion.

Definition 3. *Fix an agent a. Its **view** of a call sequence $\mathbf{c}$, written as $\mathbf{c}_a$, is a sequence of gossip situations connected by the successive calls in $\mathbf{c}$ in which agent a is involved. It is defined by induction as follows.*

[Base]

$$\epsilon_a := \mathsf{root},$$

[Step]

$$(\mathbf{c}.\mathsf{c})_a := \begin{cases} \mathbf{c}_a \xrightarrow{\mathsf{c}} \mathsf{s} & \text{if } a \in \mathsf{c} \\ \mathbf{c}_a & \text{otherwise} \end{cases}$$

where for $d \in \mathsf{A}$

$$\mathsf{s}_d := \begin{cases} \mathbf{c}.\mathsf{c}(\mathsf{root})_d & \text{if } d \in \mathsf{c} \\ \mathsf{s}'_d & \text{otherwise} \end{cases}$$

where s' is the last gossip situation in $\mathbf{c}_a$.

Intuitively, a view of agent a of a call sequence $\mathbf{c}$ is the information he acquires by means of the calls in $\mathbf{c}$ he is involved in. It consists of a sequence of gossip situations connected by the calls in which a is involved. After each such call, say ab, agent a updates the set of gossips he and b are currently familiar with.

Example 2. Let us return to Example 1. So $\mathsf{A} = \{a, b, c\}$ and we consider the call sequence (ac, bc, ac). We noticed there that it generates the following successive gossip situations starting from root:

$$A.B.C \xrightarrow{ac} AC.B.AC \xrightarrow{bc} AC.ABC.ABC \xrightarrow{ac} ABC.ABC.ABC.$$

We now compare it with the view of agent a of the call sequence (ac, bc, ac), which is

$$A.B.C \xrightarrow{ac} AC.B.AC \xrightarrow{ac} ABC.B.ABC.$$

Thus, in the final gossip situation of this view, agent b is familiar with neither the secret A nor C. However, the final gossip situation of a view does not reflect agents' knowledge. In fact, as we shall explain in Example 4, according to the semantics, after the above sequence of calls, agent a knows that agent b is familiar both with A and C. □

We now introduce for each agent a an equivalence relation $\equiv_a$ between the call sequences, defined as follows:

$$\mathbf{c} \equiv_a \mathbf{d} \text{ iff } \mathbf{c}_a = \mathbf{d}_a.$$

So according to this definition two call sequences are equivalent for agent a if his views of them are the same. The following result explains our interest in the views of call sequences.

Theorem 3 (Equivalence)**.** *For each agent a the relations $\sim_a$ and $\equiv_a$ coincide.*

So two call sequences are $\sim_a$ equivalent iff their views by agent a coincide.

Proof. Fix an agent a. We prove by induction on the sum of the lengths $|\mathbf{c}| + |\mathbf{d}|$ that

$$\mathbf{c} \sim_a \mathbf{d} \text{ iff } \mathbf{c} \equiv_a \mathbf{d}.$$

[Base] $|\mathbf{c}| + |\mathbf{d}| = 0$.

Then $\mathbf{c} = \mathbf{d} = \epsilon$, so the equivalence holds.

[Step] $|\mathbf{c}| + |\mathbf{d}| > 0$.
($\Rightarrow$) Suppose $\mathbf{c} \sim_a \mathbf{d}$. On the account of the minimality of the $\sim_a$ relation only the following three cases can arise.

Case 1. For some sequence $\mathbf{c}'$ and call c, we have $a \notin \mathsf{c}$, $\mathbf{c} = \mathbf{c}'.\mathsf{c}$ and $\mathbf{c}' \sim_a \mathbf{d}$.

By the definition we have then $\mathbf{c}_a = \mathbf{c}'_a$. By the induction hypothesis $\mathbf{c}' \equiv_a \mathbf{d}$, i.e., $\mathbf{c}'_a = \mathbf{d}_a$. Hence $\mathbf{c}_a = \mathbf{d}_a$, i.e., $\mathbf{c} \equiv_a \mathbf{d}$.

Case 2. For some sequence $\mathbf{d}'$ and a call c, we have $a \notin c$, $\mathbf{d} = \mathbf{d}'.\mathsf{c}$ and $\mathbf{c} \sim_a \mathbf{d}'$.

Analogous to Case 1.

Case 3. For some sequences $\mathbf{c}'$ and $\mathbf{d}'$ and a call c, we have $a \in c$, $\mathbf{c} = \mathbf{c}'.\mathsf{c}$, $\mathbf{d} = \mathbf{d}'.\mathsf{c}$, $\mathbf{c}' \sim_a \mathbf{d}'$ and $\mathbf{c}(\mathsf{root})_a = \mathbf{d}(\mathsf{root})_a$.

Let s_1 be the last gossip situation in $\mathbf{c}'_a$ and s'_1 the last gossip situation in $\mathbf{d}'_a$. By the definition of a view we have

$$\mathbf{c}_a = \mathbf{c}'_a \xrightarrow{\mathsf{c}} \mathsf{s}$$

and

$$\mathbf{d}_a = \mathbf{d}'_a \xrightarrow{\mathsf{c}} \mathsf{s}',$$

where for $d \in \mathsf{A}$

$$\mathsf{s}_d := \begin{cases} \mathbf{c}(\mathsf{root})_d & \text{if } d \in \mathsf{c} \\ (\mathsf{s}_1)_d & \text{otherwise} \end{cases}$$

and

$$\mathsf{s}'_d := \begin{cases} \mathbf{d}(\mathsf{root})_d & \text{if } d \in \mathsf{c} \\ (\mathsf{s}'_1)_d & \text{otherwise} \end{cases}$$

By the induction hypothesis $\mathbf{c}' \equiv_a \mathbf{d}'$, i.e., $\mathbf{c}'_a = \mathbf{d}'_a$ and consequently $\mathsf{s}_1 = \mathsf{s}'_1$. Moreover, by assumption $\mathbf{c}(\mathsf{root})_a = \mathbf{d}(\mathsf{root})_a$, i.e., $\mathsf{s}_a = \mathsf{s}'_a$. Further, the last calls in $\mathbf{c}$ and $\mathbf{d}$ are the same, say $\mathsf{c} = ab$, so $\mathsf{s}_b = \mathsf{s}_a$ and $\mathsf{s}'_b = \mathsf{s}'_a$, and hence $\mathsf{s}_b = \mathsf{s}'_b$.

This shows that $\mathsf{s} = \mathsf{s}'$. Consequently $\mathbf{c}_a = \mathbf{d}_a$, i.e., $\mathbf{c} \equiv_a \mathbf{d}$.

($\Leftarrow$) Suppose $\mathbf{c} \equiv_a \mathbf{d}$. Three cases arise.

Case 1. The last call in $\mathbf{c}$ does not involve agent a.

For some sequence $\mathbf{c}'$ and call c, we have $\mathbf{c} = \mathbf{c}'.\mathsf{c}$. By the definition of $\equiv_a$ we have $\mathbf{c}_a = \mathbf{d}_a$. Further, by definition $\mathbf{c}_a = \mathbf{c}'_a$, so $\mathbf{c}' \equiv_a \mathbf{d}$. By the induction hypothesis $\mathbf{c}' \sim_a \mathbf{d}$, so by the definition of $\sim_a$ we have $\mathbf{c} \sim_a \mathbf{d}$.

Case 2. The last call in $\mathbf{d}$ does not involve agent a.

Analogous to Case 1.

Case 3. The last calls in $\mathbf{c}$ and in $\mathbf{d}$ involve the agent a. Since the views of $\mathbf{c}$ and $\mathbf{d}$ are the same, these last calls coincide and equal some call c. For some sequences $\mathbf{c}'$ and $\mathbf{d}'$ we have $\mathbf{c} = \mathbf{c}'.\mathsf{c}$ and $\mathbf{d} = \mathbf{d}'.\mathsf{c}$.

By the definition of $\equiv_a$ we have $\mathbf{c}_a = \mathbf{d}_a$. This implies $\mathbf{c}(\mathsf{root})_a = \mathbf{d}(\mathsf{root})_a$ and $\mathbf{c}'_a = \mathbf{d}'_a$, i.e., $\mathbf{c}' \equiv_a \mathbf{d}'$. By the induction hypothesis $\mathbf{c}' \sim_a \mathbf{d}'$, so by the definition of $\sim_a$ we have $\mathbf{c} \sim_a \mathbf{d}$.

This concludes the proof. □

This alternative definition of the equivalence relation between the call sequences makes it simpler to determine various properties of our semantics. In the examples and proofs below we use the $\equiv_a$ relation instead of $\sim_a$ and repeatedly appeal to the Equivalence Theorem 3.

Given a call sequence $\mathbf{c}$, we denote by $\mathbf{c}^*$ a call sequence consisting of zero or more repetitions of $\mathbf{c}$, and by $\mathbf{c}^+$ a call sequence consisting of one or more repetitions of $\mathbf{c}$. Given two call sequences $\mathbf{c}$ and $\mathbf{d}$, we denote by $(\mathbf{c} \mid \mathbf{d})$ a call sequence that is either equal $\mathbf{c}$ or $\mathbf{d}$. Finally, for a call ab, we write $\overline{ab}$ as a shorthand for $(ab \mid ba)$.

We say that a call sequence $\mathbf{c}$ ***is of the form*** $(\overline{ab})^*.ac.(bc)^+$ if $\mathbf{c}$ is equal to $(\overline{ab})^m.ac.(bc)^n$ for some $m \geq 0$ and $n > 0$. An analogous terminology is used for other call sequences.

Example 4. Suppose that $\mathsf{A} = \{a, b, c\}$ and that $(ac, bc, ac) \equiv_a \mathbf{d}$. Recall from Example 2 that the view of agent a of the sequence (ac, bc, ac) is

$$A.B.C \xrightarrow{ac} AC.B.AC \xrightarrow{ac} ABC.B.ABC.$$

By the Equivalence Theorem 3 this is also the view of agent a of the call sequence $\mathbf{d}$. So there are precisely two calls ac in $\mathbf{d}$. Hence $\mathbf{d}$ is of the form $(\overline{bc})^*.ac.(\overline{bc})^*.ac.(\overline{bc})^*$. But the second gossip situation of the view is $AC.B.AC$, so $\mathbf{d}$ is actually of the form $ac.(\overline{bc})^*.ac.(\overline{bc})^*$. Further, the third gossip situation of the view is $ABC.B.ABC$, so we conclude that $\mathbf{d}$ is in fact of the form $ac.(\overline{bc})^+.ac.(\overline{bc})^*$.

This implies that $(\mathcal{M}, \mathbf{d}) \models F_b A$. It follows that $(\mathcal{M}, (ac, bc, ac)) \models K_a F_b A$. We conclude that it is possible that an agent, here a, knows that another agent, here b, is familiar with his (so a's) secret even though no communication took place between them. The same argument shows that $(\mathcal{M}, (ac, bc, ac)) \models K_a F_b C$, as claimed in Example 2. □

Corollary 5.

(i) Each $\sim_a$ is an equivalence relation.

(ii) For all $\mathbf{c}, \mathbf{d} \in \mathsf{C}$ and agents a, if $\mathbf{c} \sim_a \mathbf{d}$, then $\mathbf{c}(\mathsf{root})_a = \mathbf{d}(\mathsf{root})_a$.

Proof. (i) By the Equivalence Theorem 3.

(ii) By the Equivalence Theorem 3 it suffices to show that $\mathbf{c}_a = \mathbf{d}_a$ implies that $\mathbf{c}(\mathsf{root})_a = \mathbf{d}(\mathsf{root})_a$.

Let c and d be the last calls in $\mathbf{c}$ and $\mathbf{d}$, respectively, that involve agent a, i.e., such that $a \in \mathsf{c}$ and $a \in \mathsf{d}$. This means that for some $\mathbf{c}'$, $\mathbf{c}''$, $\mathbf{d}'$, $\mathbf{d}''$ we have $\mathbf{c} = \mathbf{c}'.\mathsf{c}.\mathbf{c}''$ and $\mathbf{d} = \mathbf{d}'.\mathsf{d}.\mathbf{d}''$, and neither $\mathbf{c}''$ nor $\mathbf{d}''$ contains an a-call.

By the assumption and the definition of a view we have $(\mathbf{c}'.\mathsf{c})_a = \mathbf{c}_a = \mathbf{d}_a = (\mathbf{d}'.\mathsf{d})_a$ and consequently, again the definition of a view, $\mathbf{c}'.\mathsf{c}(\mathsf{root})_a = \mathbf{d}'.\mathsf{d}(\mathsf{root})_a$. Further, $\mathbf{c}(\mathsf{root})_a = \mathbf{c}'.\mathsf{c}.\mathbf{c}''(\mathsf{root})_a = \mathbf{c}'.\mathsf{c}(\mathsf{root})_a$ and $\mathbf{d}(\mathsf{root})_a = \mathbf{d}'.\mathsf{d}.\mathbf{d}''(\mathsf{root})_a = \mathbf{d}'.\mathsf{d}(\mathsf{root})_a$, which yields the claim. □

Note that the implication in *(ii)* cannot be reversed, as $(ab, ab)(\mathsf{root})_a = (ab)(\mathsf{root})_a$ but $(ab, ab) \not\sim_a ab$.

Next we show that an immediate repetition of a call has no effect on the truth of the formulas. More precisely, the following holds.

Theorem 6 (Stuttering)**.** *Suppose that* $\mathbf{c} := \mathbf{c}_1.\mathsf{c}.\mathbf{c}_2$ *and* $\mathbf{d} := \mathbf{c}_1.\mathsf{c}.\mathsf{c}.\mathbf{c}_2$*. Then for all formulas* $\phi \in \mathcal{L}$*,* $(\mathcal{M}, \mathbf{c}) \models \phi$ *iff* $(\mathcal{M}, \mathbf{d}) \models \phi$*.*

Proof. We proceed by induction of the structure of ϕ. For the formulas of the form $F_a p$ it suffices to note that $\mathbf{c}(\mathsf{root}) = \mathbf{d}(\mathsf{root})$. The only induction step of interest is for the formulas of the form $K_a\phi$. Suppose first that $a \notin \mathsf{c}$. Then $\mathbf{c} \equiv_a \mathbf{d}$, so $(\mathcal{M}, \mathbf{c}) \models K_a\phi$ iff $(\mathcal{M}, \mathbf{d}) \models K_a\phi$.

Assume now that $a \in \mathsf{c}$. Suppose that $(\mathcal{M}, \mathbf{c}) \models K_a\phi$. Take $\mathbf{d}'$ such that $\mathbf{d} \equiv_a \mathbf{d}'$. Then $\mathbf{d}'$ is of the form $\mathbf{d}'_1.\mathsf{c}.\mathsf{c}.\mathbf{d}'_2$. Let $\mathbf{c}' := \mathbf{d}'_1.\mathsf{c}.\mathbf{d}'_2$. By the induction hypothesis $(\mathcal{M}, \mathbf{d}') \models \phi$ iff $(\mathcal{M}, \mathbf{c}') \models \phi$. Further, $\mathbf{d} \equiv_a \mathbf{d}'$ implies that $\mathbf{c} \equiv_a \mathbf{c}'$. So $(\mathcal{M}, \mathbf{c}') \models \phi$. Hence $(\mathcal{M}, \mathbf{d}') \models \phi$ and consequently $(\mathcal{M}, \mathbf{d}) \models K_a\phi$.

The proof in the other direction is analogous. □

The above result cannot be extended to a repetition of the call sequences. Indeed, we have $(\mathcal{M}, (ab, bc)) \models \neg F_a C$, and $(\mathcal{M}, (ab, bc, ab, bc)) \models F_a C$. On the other hand a monotonicity result holds for positive formulas.

Theorem 7 (Monotonicity)**.** *Suppose that* $\phi \in \mathcal{L}$ *is a formula that does not contain the* $\neg$ *symbol. Then*

$$\mathbf{c} \preceq \mathbf{d} \text{ and } (\mathcal{M}, \mathbf{c}) \models \phi \text{ implies } (\mathcal{M}, \mathbf{d}) \models \phi.$$

Proof. We proceed by induction on the structure of ϕ. The only case of interest is when ϕ is of the form $K_a\psi$. Suppose that $\mathbf{c} \preceq \mathbf{d}$ and $(\mathcal{M}, \mathbf{c}) \models \phi$. Take some call sequence $\mathbf{d}'$ such that $\mathbf{d} \equiv_a \mathbf{d}'$. Then for some call sequences $\mathbf{d}_1$ and $\mathbf{d}'_1$ such that $\mathbf{d}_1.\mathbf{d}'_1 = \mathbf{d}'$ we have $\mathbf{c} \equiv_a \mathbf{d}_1$.

We have by the assumption $(\mathcal{M}, \mathbf{d}_1) \models \psi$, so by the induction hypothesis $(\mathcal{M}, \mathbf{d}') \models \psi$. As $\mathbf{d}'$ was arbitrarily chosen we conclude that $(\mathcal{M}, \mathbf{d}) \models \phi$. □

Before we deal with the decidability matters consider the formula $K_a F_b C$ for pairwise different agents a, b, c. The following example reveals that it can be true in some subtle ways.

Example 8.
(i) First, note that a can learn that agent b is familiar with the secret C through a direct communication with b.

Indeed, we have $(\mathcal{M}, (bc, ab)) \models K_a F_b C$. Namely, the view of agent a of the sequence (bc, ab) is

$$A.B.C \xrightarrow{ab} ABC.ABC.C.$$

So if $(bc, ab) \equiv_a \mathbf{d}$, then by the reasoning analogous to the one given in Example 4 $\mathbf{d}$ is of the form $(\overline{bc})^+.ab.(\overline{bc})^*$, which implies that $(\mathcal{M}, \mathbf{d}) \models F_b C$.

(ii) Further, it is also possible that a learns that b is familiar with the secret C through a direct communication with c.

Indeed, we have $(\mathcal{M}, (bc, ac)) \models K_a F_b C$. To see this note that the view of agent a of the sequence (bc, ac) is

$$A.B.C \xrightarrow{ac} ABC.B.ABC.$$

So if $(bc, ac) \equiv_a \mathbf{d}$, then by the reasoning analogous to the one given in Example 4, $\mathbf{d}$ is of the form $(\overline{bc})^+.ac.(\overline{bc})^*$, which implies that $(\mathcal{M}, \mathbf{d}) \models F_bC$.

(iii) Also, it is possible that a learns that b is familiar with the secret C without ever communicating with b or c.

Namely, we have $(\mathcal{M}, (cd, ad, bd, ad)) \models K_aF_bC$. To see this note first that the view of agent a of the sequence (cd, ad, bd, ad) is

$$A.B.C.D \xrightarrow{ad} ACD.B.C.ACD \xrightarrow{ad} ABCD.B.C.ABCD.$$

Suppose now that $(cd, ad, bd, ad) \equiv_a \mathbf{d}$. Then both calls ad take place in $\mathbf{d}$. Assume first that a call $\overline{bc}$ does not take place in $\mathbf{d}$ before the first call ad. Then by the reasoning analogous to the one given in Example 4 $\mathbf{d}$ is of the form $(\overline{cd})^+.ad.\mathbf{d'}.(bd \mid bc.\mathbf{d'}.cd).\mathbf{d'}.ad.\mathbf{d'}$, where $\mathbf{d'} = (\overline{bc} \mid \overline{bd} \mid \overline{cd})^*$ is a call sequence with no a-calls. Intuitively, a call $\overline{cd}$ has to take place before the first call ad, so that agent a observes that agent d is only familiar with the secret C (apart of his own secret D). A call $\overline{bd}$ or a call $\overline{bc}$ and then a call $\overline{cd}$ has to take place before the second call ad, so that agent a observers that agent d is familiar with all the secrets after its second call. Note that this form of $\mathbf{d}$ implies that $(\mathcal{M}, \mathbf{d}) \models F_bC$.

If a call $\overline{bc}$ takes place in $\mathbf{d}$, in particular before the call ad, then $(\mathcal{M}, \mathbf{d}) \models F_bC$ holds directly.

(iv) In (iii) agent a learned that b is familiar with c by communicating with agent d twice. But it is also possible that a learns that b is familiar with the secret C without communicating with any agent twice.

To see this note that $(\mathcal{M}, (cd, ad, bc, ac)) \models K_aF_bC$. To see this note that the view of agent a of the sequence (cd, ad, bc, ac) is

$$A.B.C.D \xrightarrow{ad} ACD.B.C.ACD \xrightarrow{ac} ABCD.B.ABCD.ACD.$$

Suppose now that $(cd, ad, bc, ac) \equiv_a \mathbf{d}$ and assume that a call $\overline{bc}$ does not take place in $\mathbf{d}$ before the call ad. By the reasoning analogous to the one given in Example 4 $\mathbf{d}$ is of the form $(\overline{cd})^+.ad.\mathbf{d'}.(bc \mid bd.\mathbf{d'}.cd).\mathbf{d'}.ad.\mathbf{d'}$, where $\mathbf{d'} = (\overline{bc} \mid \overline{bd} \mid \overline{cd})^*$. Intuitively, between the calls ad and ac a call $\overline{bc}$ or a call $\overline{bd}$ followed by $\overline{cd}$ has to take place so that agent a observes that after the call ac agent c is familiar with all the secrets. This implies that $(\mathcal{M}, \mathbf{d}) \models F_bC$.

If a call $\overline{bc}$ takes place in $\mathbf{d}$ before the first call ad, the reasoning is similar and omitted. □

We conclude by noting that the Monotonicity Theorem 7 does not hold when we extend the call sequences to the left. Indeed, as observed in Example 8(ii), $(\mathcal{M}, (bc, ac)) \models K_aF_bC$. However, $(\mathcal{M}, (cd, bc, ac)) \models \neg K_aF_bC$, since $(\mathcal{M}, (bd, cd, ac)) \models \neg F_bC$ and $(cd, bc, ac) \equiv_a (bd, cd, ac)$.

7. Decidability of Semantics

In this section we show that the definition of semantics given in Definition 2 is decidable for the formulas from the language $\mathcal{L}_{wn}$.

Consider a call sequence $\mathbf{c}$. If for some prefix $\mathbf{c}_1.\mathsf{c}$ of $\mathbf{c}$ we have $\mathbf{c}_1(\mathsf{root}) = \mathbf{c}_1.\mathsf{c}(\mathsf{root})$, then we say that the call c is ***redundant*** in $\mathbf{c}$. First note the following observation.

Lemma 9 (Semantic Stuttering)**.** *Suppose that* $\mathbf{c} := \mathbf{c}_1.\mathsf{c}.\mathbf{c}_2$ *and* $\mathbf{d} := \mathbf{c}_1.\mathbf{c}_2$*, where* c *is redundant in* $\mathbf{c}$*. Then*

(i) $\mathbf{c}(\mathsf{root}) = \mathbf{d}(\mathsf{root})$*,*

(ii) for all formulas $\phi \in \mathcal{L}_{pr}$*,* $(\mathcal{M}, \mathbf{c}) \models \phi$ *iff* $(\mathcal{M}, \mathbf{d}) \models \phi$*.*

Proof.
(i) By the redundancy of c we have $\mathbf{c}_1(\mathsf{root}) = \mathbf{c}_1.\mathsf{c}(\mathsf{root})$, so $\mathbf{c}_1.\mathsf{c}.\mathbf{c}_2(\mathsf{root}) = \mathbf{c}_1.\mathbf{c}_2(\mathsf{root})$.

(ii) We proceed by induction on the structure of ϕ. The only case of interest is when ϕ is of the form $F_a p$. By (i) $(\mathcal{M}, \mathbf{c}) \models F_a p$ iff $p \in \mathbf{c}(\mathsf{root})_a$ iff $p \in \mathbf{d}(\mathsf{root})_a$ iff $(\mathcal{M}, \mathbf{d}) \models F_a p$. □

The following example shows that Lemma 9 does not extend to arbitrary formulas of $\mathcal{L}$.

Example 10. In the call sequence (ab, ac, bc, ab) the second call ab is redundant since

$$(ab, ac, bc, ab)(\mathsf{root}) = (ab, ac, bc)(\mathsf{root}) = ABC.ABC.ABC.$$

We have $(\mathcal{M}, (ab, ac, bc, ab)) \models K_a F_b C$, because if $(ab, ac, bc, ab) \equiv_a \mathbf{d}$ then by the reasoning analogous to the one given in Example 4, $\mathbf{d}$ is of the form $ab.ac.(\overline{bc})^+.ab.(\overline{bc})^*$. However, $(\mathcal{M}, (ab, ac, bc)) \models \neg K_a F_b C$ since $(ab, ac, bc) \equiv_a (ab, ac)$. □

Now, consider an agent a and a call sequence $\mathbf{c}$. Starting from $\mathbf{c}$ we repeatedly remove from the current call sequence a redundant call that does not involve agent a. We call each outcome of such an iteration an ***a-reduction*** of $\mathbf{c}$.

Corollary 11. *Let* $\mathbf{d}$ *be an* a*-reduction of* $\mathbf{c}$*. Then*

(i) $\mathbf{c} \equiv_a \mathbf{d}$*,*

(ii) $\mathbf{c}(\mathsf{root}) = \mathbf{d}(\mathsf{root})$*,*

(iii) for all formulas $\phi \in \mathcal{L}_{pr}$*,* $(\mathcal{M}, \mathbf{c}) \models \phi$ *iff* $(\mathcal{M}, \mathbf{d}) \models \phi$*.*

Proof.
(i) It suffices to note that a removal of a redundant call that does not involve agent a does not affect his view of the call sequence.

(ii) and (iii) By the repeated use of the Semantic Stuttering Lemma 9. □

Given an agent a we now say that a call sequence $\mathbf{c}$ is ***a-redundant free*** if no call c from $\mathbf{c}$ such that $a \notin \mathsf{c}$ is redundant in it. Clearly each a-reduction is a-redundant free.

We now prove the following crucial lemma.

Lemma 12. *For each agent* a *and a call sequence* $\mathbf{c}$ *the set of* a*-redundant free call sequences* $\mathbf{d}$ *such that* $\mathbf{c} \equiv_a \mathbf{d}$ *is finite and can be effectively constructed.*

Proof. Consider an a-redundant free call sequence $\mathbf{d}$ such that $\mathbf{c} \equiv_a \mathbf{d}$. Then $\mathbf{d}$ has the same number, say k, of a-calls as $\mathbf{c}$.

Suppose $\mathbf{d} = \mathsf{d}_1.\mathsf{d}_2.\ldots.\mathsf{d}_m$, where m is the length of $\mathbf{d}$. Associate with $\mathbf{d}$ the sequence of gossip situations $\mathbf{d}^0(\mathsf{root}), \mathbf{d}^1(\mathsf{root}), \ldots, \mathbf{d}^m(\mathsf{root})$, where $\mathbf{d}^0 = \epsilon$, and $\mathbf{d}^i = \mathsf{d}_1.\mathsf{d}_2.\ldots.\mathsf{d}_i$ for $i = 1, \ldots, m$. This sequence monotonically grows, where we interpret the inclusion relation componentwise. Moreover, for all calls d_j such that $a \notin \mathsf{d}_j$ the corresponding inclusion is strict. Consequently, m, the length of $\mathbf{d}$, is bounded by $k + |\mathsf{A}|^2$, the sum of the number of a-calls in $\mathbf{c}$ and of the total number of secrets in the gossip situation in which each agent is an expert.

But for each m there are only finitely many call sequences of length at most m. This concludes the proof. □

We can now state and prove the desired result.

Theorem 13 (Decidability of Semantics)**.** *For each call sequence* $\mathbf{c}$ *it is decidable whether for a formula* $\phi \in \mathcal{L}_{wn}$*,* $(\mathcal{M}, \mathbf{c}) \models \phi$ *holds.*

Proof. We use the definition of semantics as an algorithm and prove this by induction over the structure of the formulas. The only interesting case are formulas of the form $K_a\psi$, where $\psi \in \mathcal{L}_{pr}$. Thanks to the Equivalence Theorem 3 and Corollary 11 we can rewrite the clause for $K_a\psi$ as:

$$(\mathcal{M}, \mathbf{c}) \models K_a\psi \text{ iff } \forall \mathbf{d} \text{ s.t. } \mathbf{c} \sim_a \mathbf{d} \text{ and } \mathbf{d} \text{ is } a\text{-redundant free, } (\mathcal{M}, \mathbf{d}) \models \psi,$$

and according to Lemma 12 this definition refers to an explicitly constructed finite set of call sequences $\mathbf{d}$, so the problem is decidable. □

We now apply this result to gossip protocols. We say that a gossip protocol is ***implementable*** if an effective procedure exists that allows one to determine whether a guard is true after a sequence of calls generated by the protocol. We have the following result.

Corollary 14. *Each gossip protocol that uses as guards the formulas from* $\mathcal{L}_{wn}$ *is implementable.*

Proof. By the Decidability of Semantics 13. □

8. Decidability of Truth

Next, we show that truth definition for the formulas of the language $\mathcal{L}_{wn}$ is decidable. The key notion in our approach is the following.

Definition 4. *An* ***epistemic view*** *is a function* $\mathsf{EV}(\cdot)$ *defined on the set of call sequences such that for each call sequence* $\mathbf{c}$*,* $\mathsf{EV}(\mathbf{c})$ *is in turn a function with the domain* $\mathsf{A} \cup \{\mathsf{A}\}$ *that assigns to each agent* $a \in \mathsf{A}$ *a set of gossip situations and to the set of agents* A *a single gossip situation. It is defined by*

- *putting for each agent* $a \in \mathsf{A}$*,* $\mathsf{EV}(\mathbf{c})(a) = \{\mathbf{d}(\mathsf{root}) \mid \mathbf{c} \sim_a \mathbf{d}\}$*, and setting*
- $\mathsf{EV}(\mathbf{c})(\mathsf{A}) = \mathbf{c}(\mathsf{root})$.

So $\mathsf{EV}(\mathbf{c})(a)$ is the set of all gossip situations consistent with agent a's observations made throughout $\mathbf{c}$ and $\mathsf{EV}(\mathbf{c})(\mathsf{A})$ is the actual gossip situation after $\mathbf{c}$ takes place. Note that if $\mathbf{c} \sim_a \mathbf{d}$ then $\mathsf{EV}(\mathbf{c})(a) = \mathsf{EV}(\mathbf{d})(a)$.

Example 15. Consider a model with three agents $\mathsf{A} = \{a, b, c\}$ and let us look at all epistemic views along the call sequence (ab, ac, ab, ac).

$$
\begin{array}{l}
\mathsf{EV}(\epsilon)(\mathsf{A}) = \{A.B.C\},\\
\mathsf{EV}(\epsilon)(a) = \{A.B.C, A.BC.BC\},\\
\mathsf{EV}(\epsilon)(b) = \{A.B.C, AC.B.AC\},\\
\mathsf{EV}(\epsilon)(c) = \{A.B.C, AB.AB.C\},\\
\mathsf{EV}(ab)(\mathsf{A}) = \{AB.AB.C\},\\
\mathsf{EV}(ab)(a) = \{AB.AB.C, AB.ABC.ABC\},\\
\mathsf{EV}(ab)(b) = \{AB.AB.C, ABC.AB.ABC\},\\
\mathsf{EV}(ab)(c) = \{A.B.C, AB.AB.C\},\\
\mathsf{EV}(ab, ac)(\mathsf{A}) = \{ABC.AB.ABC\},\\
\mathsf{EV}(ab, ac)(a) = \{ABC.AB.ABC, ABC.ABC.ABC\},\\
\mathsf{EV}(ab, ac)(b) = \{AB.AB.C, ABC.AB.ABC\},\\
\mathsf{EV}(ab, ac)(c) = \{ABC.AB.ABC, ABC.ABC.ABC\},\\
\mathsf{EV}(ab, ac, ab)(\mathsf{A}) = \{ABC.ABC.ABC\},\\
\mathsf{EV}(ab, ac, ab)(a) = \{ABC.ABC.ABC\},\\
\mathsf{EV}(ab, ac, ab)(b) = \{ABC.ABC.ABC\},\\
\mathsf{EV}(ab, ac, ab)(c) = \{ABC.AB.ABC, ABC.ABC.ABC\},\\
\mathsf{EV}(ab, ac, ab, ac)(\mathsf{A}) = \{ABC.ABC.ABC\},\\
\mathsf{EV}(ab, ac, ab, ac)(a) = \{ABC.ABC.ABC\},\\
\mathsf{EV}(ab, ac, ab, ac)(b) = \{ABC.ABC.ABC\},\\
\mathsf{EV}(ab, ac, ab, ac)(c) = \{ABC.AB.ABC, ABC.ABC.ABC\}.
\end{array}
$$

The last equality holds since by the Equivalence Theorem 3, $(ab, ac, ab, ac) \sim_c \mathbf{d}$ holds iff $\mathbf{d}$ is of the form $(\overline{ab})^+.ac.(\overline{ab})^*.ac.(\overline{ab})^*$. In particular $(ab, ac, ab, ac) \sim_c (ba, ac, ac)$. We leave the checking of the other equalities to the reader. □

Consider now the set $\mathsf{EV}(\mathbf{c})(a)$ for some call sequence $\mathbf{c}$ and agent a. Even though it is defined using infinitely many call sequences, it is finite because the set of gossip situations is finite. In what follows we need a stronger observation.

Lemma 16. *For each call sequence* $\mathbf{c}$ *and agent* a *the set* $\mathsf{EV}(\mathbf{c})(a)$ *is finite and can be effectively constructed.*

Proof. Fix an agent a. By Corollary 11 and Equivalence Theorem 3 to construct the set $\mathsf{EV}(\mathbf{c})(a)$ it suffices to consider a-redundant free call sequences $\mathbf{d}$ and by Lemma 12 there are only finitely many such call sequences $\mathbf{d}$ for which $\mathbf{d} \sim_a \mathbf{c}$. □

Our interest in epistemic views stems from the following result.

Lemma 17. *Suppose that* $\mathsf{EV}(\mathbf{c}) = \mathsf{EV}(\mathbf{d})$. *Then for all formulas* $\phi \in \mathcal{L}_{wn}$, $(\mathcal{M}, \mathbf{c}) \models \phi$ *iff* $(\mathcal{M}, \mathbf{d}) \models \phi$.

So to determine whether two call sequences satisfy the same formulas of $\mathcal{L}_{wn}$ it suffices to compare their epistemic views which are finite objects.

Proof. A simple proof by induction shows that for a formula $\psi \in \mathcal{L}_{pr}$ and arbitrary call sequences $\mathbf{c}'$ and $\mathbf{d}'$,

$$\mathbf{c}'(\mathsf{root}) = \mathbf{d}'(\mathsf{root}) \text{ implies that } (\mathcal{M}, \mathbf{c}') \models \psi \text{ iff } (\mathcal{M}, \mathbf{d}') \models \psi. \tag{3}$$

Since $\mathsf{EV}(\mathbf{c})(\mathsf{A}) = \mathbf{c}(\mathsf{root})$ and $\mathsf{EV}(\mathbf{d})(\mathsf{A}) = \mathbf{d}(\mathsf{root})$, this settles the case for $\phi = F_a p$.

Now consider $\phi = K_a\psi$ where $\psi \in \mathcal{L}_{pr}$. Recall that

$$(\mathcal{M}, \mathbf{c}) \models K_a\psi \quad \text{iff} \quad \forall \mathbf{c}' \sim_a \mathbf{c},\ (\mathcal{M}, \mathbf{c}') \models \psi.$$

Due to (3) the last condition can be rewritten as

$$\forall \mathbf{c}'' \text{ for which } \exists \mathbf{c}' \text{ such that } \mathbf{c}' \sim_a \mathbf{c} \text{ and } \mathbf{c}''(\mathsf{root}) = \mathbf{c}'(\mathsf{root}),\ (\mathcal{M}, \mathbf{c}'') \models \psi.$$

Finally, due to the definition of $\mathsf{EV}(\mathbf{c})(a)$ this can be simplified to

$$\forall \mathbf{c}'' \text{ such that } \mathbf{c}''(\mathsf{root}) \in \mathsf{EV}(\mathbf{c})(a),\ (\mathcal{M}, \mathbf{c}'') \models \psi.$$

Since $\mathsf{EV}(\mathbf{c})(a) = \mathsf{EV}(\mathbf{d})(a)$, this settles the case for $\phi = K_a\psi$. The remaining cases of negation and conjunction follow directly by the induction. □

Next, we provide an inductive definition of $\mathsf{EV}(\mathbf{c}.\mathsf{c})(a)$ the importance of which will become clear in a moment.

Lemma 18. *For any call sequence* $\mathbf{c}$*, call* c*, and agent* a *such that* $a \in \mathsf{c}$

$$\mathsf{EV}(\mathbf{c}.\mathsf{c})(a) = \{\mathsf{c}(\mathsf{s}) \mid \mathsf{s} \in \mathsf{EV}(\mathbf{c})(a) \text{ and } \mathsf{c}(\mathsf{s})_a = \mathsf{c}(\mathbf{c}(\mathsf{root}))_a\}.$$

Proof. Intuitively the condition $\mathsf{c}(\mathsf{s})_a = \mathsf{c}(\mathbf{c}(\mathsf{root}))_a$ states that s is consistent with the observation agent a gets after call c is made in the gossip situation $\mathbf{c}(\mathsf{root})$.

($\subseteq$) Take $\mathsf{s}' \in \mathsf{EV}(\mathbf{c}.\mathsf{c})(a)$. By the definition of $\mathsf{EV}(\mathbf{c}.\mathsf{c})(a)$ there exists a call sequence $\mathbf{d}.\mathsf{c}$ such that $\mathbf{d}.\mathsf{c} \sim_a \mathbf{c}.\mathsf{c}$ and $\mathsf{s}' = \mathbf{d}.\mathsf{c}(\mathsf{root})$. So $\mathsf{s}' = \mathsf{c}(\mathsf{s})$, where $\mathsf{s} = \mathbf{d}(\mathsf{root})$. We also have $\mathbf{d} \sim_a \mathbf{c}$, so $\mathbf{d}(\mathsf{root}) \in \mathsf{EV}(\mathbf{c})(a)$. Moreover, $\mathsf{c}(\mathbf{d}(\mathsf{root}))_a = \mathsf{c}(\mathbf{c}(\mathsf{root}))_a$, because $\mathbf{d}.\mathsf{c} \sim_a \mathbf{c}.\mathsf{c}$.

($\supseteq$) Take $\mathsf{s}' \in \{\mathsf{c}(\mathsf{s}) \mid \mathsf{s} \in \mathsf{EV}(\mathbf{c})(a) \text{ and } \mathsf{c}(\mathsf{s})_a = \mathsf{c}(\mathbf{c}(\mathsf{root}))_a\}$. So for some gossip situation s we have $\mathsf{s}' = \mathsf{c}(\mathsf{s})$, $\mathsf{s} \in \mathsf{EV}(\mathbf{c})(a)$ and $\mathsf{c}(\mathsf{s})_a = \mathsf{c}(\mathbf{c}(\mathsf{root}))_a$. The second fact implies that there exists a call sequence $\mathbf{d}$ such that $\mathbf{d} \sim_a \mathbf{c}$ and $\mathsf{s} = \mathbf{d}(\mathsf{root})$. Now, this and the third fact imply that $\mathbf{d}.\mathsf{c} \sim_a \mathbf{c}.\mathsf{c}$. So $\mathbf{d}.\mathsf{c}(\mathsf{root}) \in \mathsf{EV}(\mathbf{c}.\mathsf{c})(a)$. Consequently also $\mathsf{s}' \in \mathsf{EV}(\mathbf{c}.\mathsf{c})(a)$, since $\mathsf{s}' = \mathsf{c}(\mathsf{s}) = \mathbf{d}.\mathsf{c}(\mathsf{root})$. □

This brings us to the following important conclusion that the function $\mathsf{EV}(\mathbf{c}.\mathsf{c})$ can be computed using $\mathsf{EV}(\mathbf{c})$ and c only, i.e., without referring to $\mathbf{c}$. Denote the set of epistemic views by $\widetilde{\mathsf{EV}}$ and recall that C denotes the set of calls.

Corollary 19. *There exists a function* $f : \widetilde{\mathsf{EV}} \times \mathsf{C} \to \widetilde{\mathsf{EV}}$ *such that for any call sequence* $\mathbf{c}$ *and call* c

$$\mathsf{EV}(\mathbf{c}.\mathsf{c}) = f(\mathsf{EV}(\mathbf{c}), \mathsf{c}).$$

Proof. By the definition of $\sim_a$ we have $\mathsf{EV}(\mathbf{c}.\mathsf{c})(a) = \mathsf{EV}(\mathbf{c})(a)$ if $a \notin \mathsf{c}$. Also $\mathsf{EV}(\mathbf{c}.\mathsf{c})(\mathsf{A}) = \mathsf{c}(\mathsf{EV}(\mathbf{c})(\mathsf{A}))$. This in conjunction with the above lemma implies the claim. □

A crucial role in the subsequent considerations will be played by the following notion. Consider a call sequence $\mathbf{c}$. If for some prefix $\mathbf{c}_1.\mathbf{c}_2$ of $\mathbf{c}$, where $\mathbf{c}_2$ is non-empty, we have $\mathsf{EV}(\mathbf{c}_1) = \mathsf{EV}(\mathbf{c}_1.\mathbf{c}_2)$, then we say that the call subsequence $\mathbf{c}_2$ is ***epistemically redundant*** in $\mathbf{c}$ and that $\mathbf{c}$ is ***epistemically redundant***. We say that $\mathbf{c}$ is ***epistemically non-redundant*** if it is not epistemically redundant.

Equivalently, a call sequence $\mathsf{c}_1.\mathsf{c}_2.\ldots.\mathsf{c}_k$ is epistemically redundant if the set

$$\{\mathsf{EV}(\mathsf{c}_1.\mathsf{c}_2.\ldots.\mathsf{c}_i) \mid i \in \{1,\ldots,k\}\}$$

has less than k elements and is epistemically non-redundant if it has k elements. In other words, in an epistemically non-redundant call sequence the successive epistemic views along the sequence are all different.

Example 20. Let us return to Example 15. We defined there the $\mathsf{EV}(\cdot)$ functions for five call sequences ϵ, ab, (ab, ac), (ab, ac, ab), and (ab, ac, ab, ac). Note that

$$\mathsf{EV}(\epsilon) \neq \mathsf{EV}(ab) \neq \mathsf{EV}(ab, ac) \neq \mathsf{EV}(ab, ac, ab) = \mathsf{EV}(ab, ac, ab, ac).$$

This shows that the second call ac in the call sequence (ab, ac, ab, ac) is epistemically redundant and no other call is epistemically redundant in this call sequence. Also, the call sequences ϵ, ab, (ab, ac) and (ab, ac, ab) are all epistemically non-redundant, while (ab, ac, ab, ac) is epistemically redundant. □

The notions of a redundant call and of an epistemically redundant call differ. Indeed, we noted in Example 10 that in the call sequence (ab, ac, bc, ab) the second call ab is redundant. Further, we also noted there that $(\mathcal{M}, (ab, ac, bc, ab)) \models K_a F_b C$ and $(\mathcal{M}, (ab, ac, bc)) \models \neg K_a F_b C$. So by Lemma 17 the second call ab is not epistemically redundant in the call sequence (ab, ac, bc, ab).

We now show a counterpart of the Semantic Stuttering Lemma 9 for epistemic views.

Lemma 21 (Epistemic Stuttering)**.** *Suppose that* $\mathbf{c} := \mathbf{c}_1.\mathbf{c}_2.\mathbf{c}_3$ *and* $\mathbf{d} := \mathbf{c}_1.\mathbf{c}_3$*, where* $\mathbf{c}_2$ *is epistemically redundant in* $\mathbf{c}$*. Then* $\mathsf{EV}(\mathbf{c}) = \mathsf{EV}(\mathbf{d})$*.*

Proof. Let $\mathbf{c}_3 = \mathsf{c}_1.\mathsf{c}_2.\ldots.\mathsf{c}_k$. First note that thanks to Corollary 19 we have $\mathsf{EV}(\mathbf{c}_1.\mathbf{c}_2.\mathsf{c}_1) = \mathsf{EV}(\mathbf{c}_1.\mathsf{c}_1)$, since $\mathsf{EV}(\mathbf{c}_1.\mathbf{c}_2.\mathsf{c}_1) = f(\mathsf{EV}(\mathbf{c}_1.\mathbf{c}_2), \mathsf{c}_1) = f(\mathsf{EV}(\mathbf{c}_1), \mathsf{c}_1) = \mathsf{EV}(\mathbf{c}_1.\mathsf{c}_1)$ due to the epistemic redundancy of $\mathbf{c}_2$ in $\mathbf{c}$. Repeating this argument for all $i \in \{1,\ldots,k\}$ we get that $\mathsf{EV}(\mathbf{c}_1.\mathbf{c}_2.\mathsf{c}_1.\mathsf{c}_2.\ldots.\mathsf{c}_i) = \mathsf{EV}(\mathbf{c}_1.\mathsf{c}_1.\mathsf{c}_2.\ldots.\mathsf{c}_i)$.

In particular $\mathsf{EV}(\mathbf{c}) = \mathsf{EV}(\mathbf{d})$. □

Corollary 22. *For every call sequence* $\mathbf{c}$ *there exists an epistemically non-redundant call sequence* $\mathbf{d}$ *such that for all formulas* $\phi \in \mathcal{L}_{wn}$*,* $(\mathcal{M}, \mathbf{c}) \models \phi$ *iff* $(\mathcal{M}, \mathbf{d}) \models \phi$*.*

Proof. By the repeated use of the Epistemic Stuttering Lemma 21 and Lemma 17. □

Next, we prove the following crucial lemma.

Lemma 23. *The set of epistemically non-redundant call sequences is finite.*

Proof. Recall that each epistemic view is a function from $\mathsf{A} \cup \{\mathsf{A}\}$ to the set of functions from A to $2^{|\mathsf{P}|}$ (this is an overestimation because for elements of A this set has only one element). There are $k = 2^{(|\mathsf{A}|+1)\cdot 2^{|\mathsf{A}|\cdot|\mathsf{P}|}}$ such functions, so any call sequence longer than k has an epistemically redundant call subsequence. But there are only finitely many call sequences of length at most k. This concludes the proof. □

This brings us to the announced result.

Theorem 24 (Decidability of Truth)**.** *For every gossip model and formula $\phi \in \mathcal{L}_{wn}$, it is decidable whether $\mathcal{M} \models \phi$ holds.*

Proof. Recall that $\mathcal{M} \models \phi$ iff $\forall \mathbf{c}\ (\mathcal{M}, \mathbf{c}) \models \phi$. Thanks to Corollary 22 we can rewrite the latter as

$$\forall \mathbf{c} \text{ s.t. } \mathbf{c} \text{ is epistemically non-redundant, } (\mathcal{M}, \mathbf{c}) \models \phi.$$

But according to Lemma 23 there are only finitely many epistemically non-redundant call sequences and by Lemma 16 their set can be explicitly constructed. □

As an easy consequence we obtain the following result.

Corollary 25. *It is decidable to determine whether a given gossip situation can be an outcome of a call sequence.*

Proof. Let $\phi(\mathsf{s})$ be the following formula of $\mathcal{L}_{wn}$ that encodes the gossip situation s:

$$\phi(\mathsf{s}) = \bigwedge_{a \in \mathsf{A}} \Big(\bigwedge_{B \in Q_a} F_a B \wedge \bigwedge_{B \notin Q_a} \neg F_a B \Big).$$

Then $\exists \mathbf{c}(\mathbf{c}(\mathsf{root}) = \mathsf{s})$ iff $\exists \mathbf{c}((\mathcal{M}, \mathbf{c}) \models \phi(\mathsf{s}))$ iff $\neg(\mathcal{M} \models \neg\phi(\mathsf{s}))$. □

(Apt et al., 2017) established that this problem is in fact NP-complete.

9. Decidability of Partial Correctness and Termination

We now explain how to apply the results of the previous section to show decidability of two crucial properties of a gossip protocol: partial correctness and termination.

We begin by establishing monotonicity of gossip situations and epistemic views with respect to the call sequence extensions. Intuitively, we claim that as the call sequence gets longer each agent acquires more information. This can be seen as a counterpart of the Monotonicity Theorem 7. First we need to define suitable partial orderings $\leq_{\mathsf{G}}$ and $\leq_{\mathsf{EV}}$ over gossip situations and epistemic views, respectively.

Definition 5. *For any two gossip situations s, s' we write $\mathsf{s} \leq_{\mathsf{G}} \mathsf{s}'$ if for all $a \in \mathsf{A}$ we have $\mathsf{s}_a \subseteq \mathsf{s}'_a$.*

Note 26. *For all call sequences $\mathbf{c}$ and $\mathbf{d}$ such that $\mathbf{c} \preceq \mathbf{d}$ we have $\mathbf{c}(\mathsf{root}) \leq_{\mathsf{G}} \mathbf{d}(\mathsf{root})$.*

Proof. For any gossip situation s and call c we have by definition $\mathsf{s} \leq_{\mathsf{G}} \mathsf{c}(\mathsf{s})$. By induction this implies that for any call sequence $\mathbf{c}'$ we have $\mathsf{s} \leq_{\mathsf{G}} \mathbf{c}'(\mathsf{s})$. Now $\mathbf{c} \preceq \mathbf{d}$ implies that $\mathbf{d} = \mathbf{c}.\mathbf{c}'$ for some $\mathbf{c}'$. Therefore, $\mathbf{c}(\mathsf{root}) \leq_{\mathsf{G}} \mathbf{c}'(\mathbf{c}(\mathsf{root})) = \mathbf{d}(\mathsf{root})$. □

Definition 6. *For any two epistemic views $V, V' \in \widetilde{\mathsf{EV}}$ we write $V \leq_{\mathsf{EV}} V'$ if for all $a \in \mathsf{A}$ there exists $X \subseteq V(a)$ and an surjective (onto) function $g : X \to V'(a)$ such that for all $\mathsf{s} \in X$ we have $\mathsf{s} \leq_{\mathsf{G}} g(\mathsf{s})$.*

Lemma 27. *$\leq_{\mathsf{EV}}$ is a partial order.*

Proof.
(Reflexivity) For any epistemic view V, we have $V \leq_{\mathsf{EV}} V$, because for each $a \in \mathsf{A}$ we can pick $V(a)$ as X and the identity function on $V(a)$ as g.

(Transitivity) Suppose V, V', V'' are three epistemic views such that $V \leq_{\mathsf{EV}} V'$ and $V' \leq_{\mathsf{EV}} V''$. Then, from the definition of $\leq_{\mathsf{EV}}$, for any $a \in \mathsf{A}$ there exist $X \subseteq V(a)$, $Y \subseteq V'(a)$, and surjective functions $g : X \to V'(a)$ and $h : Y \to V''(a)$. Let $Z = \{\mathsf{s} \in X \mid g(\mathsf{s}) \in Y\}$. Note that $g|_Z : Z \to Y$, i.e., the restriction of g to Z, is surjective. The composition $g|_Z \circ h : Z \to V''(a)$ is also surjective and for any gossip situation $\mathsf{s} \in Z$ the following holds $\mathsf{s} \leq_{\mathsf{G}} g|_Z(\mathsf{s}) \leq_{\mathsf{G}} h(g|_Z(\mathsf{s})) = (g|_Z \circ h)(\mathsf{s})$.

(Antisymmetry) Suppose V, V' are two epistemic views such that $V \leq_{\mathsf{EV}} V'$ and $V' \leq_{\mathsf{EV}} V$. Then, from the definition of $\leq_{\mathsf{EV}}$, for any $a \in \mathsf{A}$ there exist $X \subseteq V(a)$, $Y \subseteq V'(a)$, and surjective functions $g : X \to V'(a)$ and $h : Y \to V(a)$. Let $Z = \{\mathsf{s} \in X \mid g(\mathsf{s}) \in Y\}$. Note that $g|_Z : Z \to Y$, i.e., the restriction of g to Z, is surjective. Moreover, $g|_Z \circ h : Z \to V(a)$ is also surjective, and because $Z \subseteq V(a)$ is finite, $Z = V(a)$, $g|_Z = g$, and $g \circ h$ is a permutation on $V(a)$. Similarly we can show that $Y = V'(a)$. Since $(g \circ h)$ is a permutation on a finite set, there exists k such that $(g \circ h)^k$ is the identity function on $V(a)$. Note that for any $\mathsf{s} \in V(a)$, we have $\mathsf{s} \leq_{\mathsf{G}} (g \circ h)(\mathsf{s})$, because $\mathsf{s} \leq_{\mathsf{G}} g(\mathsf{s}) \leq_{\mathsf{G}} h(g(\mathsf{s}))$. Now consider the sequence $\mathsf{s} \leq_{\mathsf{G}} (g \circ h)(\mathsf{s}) \leq_{\mathsf{G}} (g \circ h)^2(\mathsf{s}) \leq_{\mathsf{G}} \ldots \leq_{\mathsf{G}} (g \circ h)^k(\mathsf{s}) = \mathsf{s}$. In fact, all of the elements in this sequence have to be the same, because $\leq_{\mathsf{G}}$ is a partial order. In particular, this shows that $(g \circ h)(\mathsf{s}) = \mathsf{s}$. Therefore, $g \circ h$ is the identity function on $V(a)$. Now, for any $\mathsf{s} \in V(a)$ we have that $\mathsf{s} \leq_{\mathsf{G}} g(\mathsf{s}) \leq_{\mathsf{G}} h(g(\mathsf{s})) = (g \circ h)(\mathsf{s}) = \mathsf{s}$, so g is the identity function as well. This shows that $V(a) = V'(a)$ for all $a \in \mathsf{A}$. □

The next lemma formalizes the intuition that epistemic information grows along a call sequence.

Lemma 28. *For all two call sequences such that $\mathbf{c} \preceq \mathbf{d}$ we have $\mathsf{EV}(\mathbf{c}) \leq_{\mathsf{EV}} \mathsf{EV}(\mathbf{d})$.*

Proof. Let $\mathbf{d} = \mathbf{c}.\mathbf{c}'$. Take $a \in \mathsf{A}$. By a repeated application of Lemma 18 we get that $\mathsf{EV}(\mathbf{c}.\mathbf{c}')(a) = \{\mathbf{c}'(\mathsf{s}) \mid \mathsf{s} \in \mathsf{EV}(\mathbf{c})(a) \text{ and } \forall \mathbf{c}'' \preceq \mathbf{c}' \;\; \mathbf{c}''(\mathsf{s})_a = \mathbf{c}''(\mathbf{c}(\mathsf{root}))_a\}$. It suffices then to pick $X = \{\mathsf{s} \in \mathsf{EV}(\mathbf{c})(a) \mid \forall \mathbf{c}'' \preceq \mathbf{c}' \;\; \mathbf{c}''(\mathsf{s})_a = \mathbf{c}''(\mathbf{c}(\mathsf{root}))_a\}$ and set $g(\mathsf{s}) = \mathbf{c}'(\mathsf{s})$ for all $\mathsf{s} \in X$. It is easy to check that such $g : X \to \mathsf{EV}(\mathbf{d})$ is surjective, so $\mathsf{EV}(\mathbf{c}) \leq_{\mathsf{EV}} \mathsf{EV}(\mathbf{d})$. □

Recall that a call sequence $\mathbf{c}$ is epistemically redundant if a prefix $\mathbf{c}_1.\mathbf{c}_2$ of it exists such that $\mathsf{EV}(\mathbf{c}_1) = \mathsf{EV}(\mathbf{c}_1.\mathbf{c}_2)$. Using the above lemma we can now draw a stronger conclusion.

Lemma 29. *Suppose that $\mathbf{c}$ is epistemically redundant. Then a prefix $\mathbf{c}_1.\mathsf{c}$ of it exists such that $\mathbf{c}_1$ is epistemically non-redundant and $\mathsf{EV}(\mathbf{c}_1.\mathsf{c}) = \mathsf{EV}(\mathbf{c}_1)$.*

Proof. Let $\mathbf{c}_1.\mathbf{c}_2$ be the shortest prefix of $\mathbf{c}$ such that $\mathsf{EV}(\mathbf{c}_1) = \mathsf{EV}(\mathbf{c}_1.\mathbf{c}_2)$. Then $\mathbf{c}_1$ is epistemically non-redundant. Let $\mathbf{c}_2 = \mathsf{c}_1.\ldots.\mathsf{c}_l$. By Lemma 28 we have $\mathsf{EV}(\mathbf{c}_1) \leq_{\mathsf{G}} \mathsf{EV}(\mathbf{c}_1.\mathsf{c}_1) \leq_{\mathsf{G}} \mathsf{EV}(\mathbf{c}_1.\mathsf{c}_1.\mathsf{c}_2) \leq_{\mathsf{G}} \ldots \leq_{\mathsf{G}} \mathsf{EV}(\mathbf{c}_1.\mathsf{c}_1.\mathsf{c}_2.\ldots.\mathsf{c}_l) = \mathsf{EV}(\mathbf{c}_1.\mathbf{c}_2) = \mathsf{EV}(\mathbf{c}_1)$. Since $\leq_{\mathsf{G}}$ is a partial order, $\mathsf{EV}(\mathbf{c}_1.\mathsf{c}_1) = \mathsf{EV}(\mathbf{c}_1)$ holds. □

Finally, the following lemma allows us to modify computations using the notion of epistemic redundancy. For convenience we identify here and below a computation with the sequence of calls it generates.

Lemma 30. *Suppose that* $\overline{\mathsf{c}} = \mathsf{c}_1.\mathsf{c}_2.\ldots$ *is a (possibly infinite) computation of a gossip protocol* P *such that a call* c_i *is epistemically redundant in the prefix* $\mathsf{c}_1.\ldots.\mathsf{c}_i$*. Then* $\overline{\mathsf{c}}$ *with the call* c_i *removed is also a computation of* P*.*

Proof. By definition for every $k \geq i$ the call c_i is epistemically redundant in $\mathsf{c}_1.\ldots.\mathsf{c}_k$, so by Lemma 17 for every $k \geq i$ we have $\mathsf{EV}(\mathsf{c}_1.\ldots.\mathsf{c}_k) = \mathsf{EV}(\mathsf{c}_1.\ldots.\mathsf{c}_{i-1}.\mathsf{c}_{i+1}.\ldots.\mathsf{c}_k)$. Thus by the Epistemic Stuttering Lemma 21 for all formulas $\phi \in \mathcal{L}_{wn}$

$$(\mathcal{M}, \mathsf{c}_1.\ldots.\mathsf{c}_k) \models \phi \text{ iff } (\mathcal{M}, \mathsf{c}_1.\ldots.\mathsf{c}_{i-1}.\mathsf{c}_{i+1}.\ldots.\mathsf{c}_k) \models \phi.$$

This implies the claim. □

We can now establish the desired results.

Theorem 31 (Decidability of Partial Correctness)**.** *Partial correctness of gossip protocols that use as guards the formulas from* $\mathcal{L}_{wn}$ *is decidable.*

Proof. Fix a gossip protocol P. We construct in a top down fashion the subtree $\mathcal{T}$ of the computation tree for P so that all nodes in are $\mathcal{T}$ correspond to epistemically non-redundant call sequences. To this end consider an epistemically non-redundant node $\mathbf{c}$ and suppose that for some rule $\psi_j^a \to \mathsf{c}_j^a$ we have $(\mathcal{M}, \mathbf{c}) \models \psi_j^a$. Then we add the call sequence $\mathbf{c}.\mathsf{c}_j^a$ as a direct descendant of $\mathbf{c}$ only if it is epistemically non-redundant.

Consider now a leaf $\mathbf{c}$ of the computation tree for P. By a repeated use of Lemma 30 we can transform $\mathbf{c}$ into a leaf $\mathbf{d}$ of the computation tree for P that is epistemically non-redundant, and thus is also a leaf of $\mathcal{T}$, and moreover is such that $\mathsf{EV}(\mathbf{c}) = \mathsf{EV}(\mathbf{d})$. By Lemma 17 the condition (1) is true after $\mathbf{c}$ iff it is true after $\mathbf{d}$.

By the above two observations P is partially correct iff the condition (1) is true after every call sequence corresponding to a leaf of $\mathcal{T}$. But by Lemma 23 $\mathcal{T}$ is finite and by Lemma 16 it can be effectively constructed. So the desired conclusion follows by the Decidability of Semantics Theorem 13. □

Below by c^ω mean the infinite call sequence consisting of the infinite repetition of the call c.

Theorem 32 (Decidability of Termination)**.** *Given a gossip protocol that uses as guards the formulas from* $\mathcal{L}_{wn}$ *it is decidable to determine whether it always terminates.*

Proof. We first prove that a gossip protocol may fail to terminate iff it can generate a call sequence $\mathbf{c}.\mathsf{c}$ such that $\mathbf{c}$ is epistemically non-redundant and $\mathsf{EV}(\mathbf{c}.\mathsf{c}) = \mathsf{EV}(\mathbf{c})$.

($\Rightarrow$) Let $\overline{\mathbf{c}}$ be an infinite sequence of calls generated by the protocol. There are only finitely many epistemic views, so some prefix $\mathbf{c}$ of $\overline{\mathbf{c}}$ is epistemically redundant. The claim now follows by Lemma 29.

($\Leftarrow$) Suppose that the protocol generates a sequence of calls $\mathbf{c}.\mathsf{c}$ such that $\mathbf{c}$ is epistemically non-redundant and $\mathsf{EV}(\mathbf{c}.\mathsf{c}) = \mathsf{EV}(\mathbf{c})$.

Let ϕ be the guard associated with the call c, i.e., $\phi \to \mathsf{c}$ is a rule used in the considered protocol. By assumption $(\mathcal{M}, \mathbf{c}) \models \phi$, so by Lemma 17 $(\mathcal{M}, \mathbf{c}.\mathsf{c}) \models \phi$. Hence by the repeated use of the Stuttering Theorem 6, for all $i \geq 1$, $(\mathcal{M}, \mathbf{c}.\mathsf{c}^i) \models \phi$. Consequently, $\mathbf{c}.\mathsf{c}^\omega$ is an infinite sequence of calls that can be generated by the protocol.

The above equivalence shows that determining whether the protocol always terminates is equivalent to checking that it cannot generate a call sequence $\mathbf{c}.\mathsf{c}$ such that $\mathbf{c}$ is epistemically non-redundant and $\mathsf{EV}(\mathbf{c}.\mathsf{c}) = \mathsf{EV}(\mathbf{c})$.

But given a call sequence, by the Decidability of Semantics Theorem 13, it is decidable to determine whether it can be generated by the protocol and by Lemma 16 it is decidable to determine whether a call sequence is epistemically non-redundant. Further, by Lemma 23 there are only finitely many epistemically non-redundant call sequences, so the claim follows. □

10. Decidability of Fair Termination

In this section we modify the approach of the previous section and show that both forms of fair termination introduced in Section 4 are decidable. First, let us clarify various forms of termination. We say that a gossip protocol ***can terminate*** if some computation of it is finite. Obviously the following implications hold for every gossip protocol P:

P terminates $\to$ P agent-fairly terminates $\to$ P rule-fairly terminates $\to$ P can terminate.

We now illustrate by means of examples that none of these implications can be reversed, even for partially correct gossip protocols. The first example exhibits a partially correct gossip protocol that may not terminate but does agent-fairly terminate.

Example 33. Let $\mathsf{A} = \{0, \ldots, k-1\}$, where $k \geq 3$. Define $i \oplus 1 = (i+1) \bmod k$ and $i \ominus 1 = (i-1) \bmod k$.

Consider a gossip protocol with the following program for each agent $i \in \mathsf{A}$:

$$*[\neg \bigwedge_{A \in \mathsf{P}} F_i A \to (i, i \oplus 1)].$$

Informally, the agents form a directed ring. Agent i calls his successor in the ring, agent $i \oplus 1$, if i is not an expert.

This gossip protocol is partially correct since its exit condition states that each agent is an expert. However, it does not terminate. Indeed, the call $(0, 1)$ can be infinitely repeated.

On the other hand this gossip protocol agent-fairly terminates. Suppose otherwise. Consider an infinite agent-fair computation ξ. We say that an agent i *becomes an expert in* ξ if for some element $\mathbf{c}$ of ξ we have $(\mathcal{M}, \mathbf{c}) \models \bigwedge_{A \in \mathsf{P}} F_i A$.

We first show that some agent becomes an expert in ξ. Indeed, otherwise by agent-fairness each agent infinitely often calls in ξ his successor. So for every agent i a sequence of calls $(i \oplus 1, i \oplus 2), (i \oplus 2, i \oplus 3), \ldots, (i \ominus 1, i)$, possibly interspersed with other calls, exists in ξ. After the last call agent i becomes an expert in ξ, which is a contradiction.

Suppose now that some agent i becomes an expert in ξ. Then also agent $i \ominus 1$ becomes an expert in ξ. Indeed, otherwise by agent-fairness agent $i \ominus 1$ infinitely often calls agent i and eventually, by the Monotonicity Theorem 7, he does become an expert in ξ.

We conclude that every agent becomes an expert in ξ. Again by the Monotonicity Theorem 7 the exit condition of the protocol is true after some element of ξ. This contradicts the fact that ξ is infinite. □

In the above gossip protocol each agent has just one rule, so agent-fairness and rule-fairness coincide. The next example shows that rule-fair termination and agent-fair termination may differ for partially correct protocols.

Example 34. Consider a gossip protocol with the following program for each agent $a \in \mathsf{A}$:

$$*[[]_{b\in\mathsf{A}} \neg \bigwedge_{C\in\mathsf{P}} F_a C \to ab].$$

Intuitively, agent a can call any other agent as long as a is not an expert. This protocol is partially correct, since the implication (2) is clearly true. However, it does not agent-fairly terminate when there are more than 3 agents.

Indeed, suppose that $|\mathsf{A}| \geq 4$. Partition A into two groups, each consisting of at least two agents, say $\{a_1, \ldots, a_k\}$ and $\{b_1, \ldots, b_m\}$, where $k, m \geq 1$. Then

$$(a_1, a_2), (a_2, a_3), \ldots, (a_k, a_1), (b_1, b_2), (b_2, b_3), \ldots, (b_m, b_1), (a_1, a_2), \ldots$$

is a sequence of calls in an infinite agent-fair computation of this protocol. Indeed, in this sequence all agents are infinitely often selected. Further, each agent learns only the secrets of the agents in its own group. So prior to each call in the above sequence no agent is an expert and consequently this sequence corresponds to a legal computation.

On the other hand, this protocol rule-fairly terminates. Indeed, consider an infinite computation χ. Some agent, say a, is then infinitely selected in χ, so it never becomes an expert and hence by the form of the protocol all the rules of a are always enabled. In χ agent a never becomes familiar with the secret of some agent, say b. So the rule $\neg \bigwedge_{C\in\mathsf{P}} F_a C \to ab$ is never selected in χ. Thus χ is not rule-fair. □

Finally, we exhibit a partially correct gossip protocol that can terminate but does not rule-fairly terminate.

Example 35. This example presents a common situation in networking where each local network has a designed gateway node, which is the only one able to communicate outside of the network (e.g., using a different network communication protocol). Here, we are going to assume just two such local networks of agents $\mathsf{A}_1 = \{a_0, a_1, \ldots, a_k\}$ and $\mathsf{A}_2 = \{b_0, \ldots, b_m\}$, where $k, m \geq 1$. So we have $\mathsf{A} = \mathsf{A}_1 \cup \mathsf{A}_2$, and a_0, b_0 are two special 'gateway' agents. Consider the gossip protocol with the following program for each agent $a \in \mathsf{A}_i$, where $i \in \{1, 2\}$:

$$*[[]_{b\in\mathsf{A}_i} \neg \bigwedge_{C\in\mathsf{P}} F_a C \to ab]$$

Intuitively, this states that agents can only directly call anyone within their own group A_i as long they are not experts. Agent a_0 has the following additional rule: $\neg F_{a_0} B_0 \to a_0 b_0$ and agent b_0 has the following additional rule: $\neg F_{b_0} A_0 \to b_0 a_0$. Intuitively, these two rules state that agents a_0 and b_0 can communicate with each other if they do not know each other secrets.

This protocol is obviously partially correct, because the implication (2) is clearly true. Further, it can terminate, as the following call sequence shows:

1. each agent in A_1 calls a_0,
2. each agent in A_2 calls b_0,
3. agent a_0 calls agent b_0,
4. each agent in A_1 calls a_0,
5. each agent in A_2 calls b_0.

The precise order of calls within each step does not matter. Notice that in step 3 all secrets from the A_1 network are passed to the A_2 network and *vice versa*. Then in the last two steps all these secrets are propagated to every agent.

Finally, it is easily to see that this protocol may not rule-fairly terminate. Indeed, let the first call be between the agents a_0 and b_0. Note that from that point on no communication between the networks A_1 and A_2 can take place. Therefore no agent will ever become an expert (for example, no agent in A_1 will ever learn the secret B_1) and agents will continue to call each other within their own network, even if one ensures that the computation is rule-fair. □

To establish decidability of both forms of fair termination we shall rely on the results established in the previous section. We start with the rule-fair termination.

Theorem 36 (Decidability of Rule-Fair Termination)**.** *Given a gossip protocol that uses as guards the formulas from $\mathcal{L}_{wn}$ it is decidable to determine whether it rule-fairly terminates.*

Proof. We first show that a gossip protocol does not rule-fairly terminate iff it can generate an epistemically non-redundant call sequence $\mathbf{c}$ such that for every call c, which is part of an enabled rule after the call sequence $\mathbf{c}$, we have that $\mathsf{EV}(\mathbf{c}.\mathsf{c}) = \mathsf{EV}(\mathbf{c})$.

($\Rightarrow$) Consider an infinite rule-fair computation $\overline{\mathbf{d}} = \mathsf{d}_1.\mathsf{d}_2\ldots$ of the considered gossip protocol. By Lemma 28 the sequence $\mathsf{EV}(\mathsf{d}_1)$, $\mathsf{EV}(\mathsf{d}_1.\mathsf{d}_2), \ldots$, is weakly increasing w.r.t. the partial order $\leq_{\mathsf{EV}}$. As there are only finitely many epistemic views, at some point this sequence stabilizes, i.e., for some l we have $\mathsf{EV}(\mathsf{d}_1.\ldots.\mathsf{d}_l) = \mathsf{EV}(\mathsf{d}_1.\ldots.\mathsf{d}_l.\mathsf{d}_{l+1}.\ldots.\mathsf{d}_{l+i})$ for all $i > 0$. Pick the smallest such l and let $\mathbf{d} = \mathsf{d}_1.\ldots.\mathsf{d}_l$.

By Lemma 30 we can repeatedly remove the epistemically redundant calls from $\mathbf{d}$ without destroying the property that it is a prefix of an infinite computation. The resulting call sequence $\mathbf{c} = \mathsf{c}_1.\ldots.\mathsf{c}_k$ is epistemically non-redundant and the resulting infinite computation $\overline{\mathbf{c}} = \mathsf{c}_1.\mathsf{c}_2\ldots$ of the protocol is rule-fair. Further, by the above choice of l and the Epistemic Stuttering Lemma 21 $\mathsf{EV}(\mathbf{c}) = \mathsf{EV}(\mathbf{c}.\mathsf{c}_{k+1}.\ldots.\mathsf{c}_{k+i})$ for all $i > 0$.

Take a rule $\psi \to \mathsf{c}$ that is enabled after $\mathbf{c}$, i.e., such that $(\mathcal{M}, \mathbf{c}) \models \psi$. By Lemma 17 and the choice of $\mathbf{c}$, this rule is enabled after each call sequence $\mathbf{c}.\mathsf{c}_{k+1}.\ldots.\mathsf{c}_{k+i}$, where $i > 0$, that is, it is enabled infinitely often. By the rule-fairness of $\overline{\mathbf{c}}$ this rule $\psi \to \mathsf{c}$ is infinitely often selected in it. So for some $i > 1$ we have $\mathsf{c} = \mathsf{c}_{k+i}$.

By the choice of k the call sequence $\mathsf{c}_{k+1}.\ldots.\mathsf{c}_{k+i-1}$ is epistemically redundant in the call sequence $\mathsf{c}_1.\ldots.\mathsf{c}_{k+i}$, so by the above equality and the Epistemic Stuttering Lemma 21:

$$\mathsf{EV}(\mathsf{c}_1.\ldots.\mathsf{c}_k) = \mathsf{EV}(\mathsf{c}_1.\ldots.\mathsf{c}_k.\mathsf{c}_{k+1}.\ldots.\mathsf{c}_{k+i}) = \mathsf{EV}(\mathsf{c}_1.\ldots.\mathsf{c}_k.\mathsf{c}_{k+i}),$$

i.e., $\mathsf{EV}(\mathbf{c}.\mathsf{c}) = \mathsf{EV}(\mathbf{c})$ as required.
($\Leftarrow$) Suppose that the protocol generates a sequence of calls $\mathbf{c}$ such that $\mathbf{c}$ is epistemically non-redundant and $\mathsf{EV}(\mathbf{c}.\mathsf{c}) = \mathsf{EV}(\mathbf{c})$ for every call c which is part of a enabled rule after the call sequence $\mathbf{c}$ takes place.

Let $R = \{\phi_1 \to \mathsf{c}_1, \phi_2 \to \mathsf{c}_2, \ldots, \phi_k \to \mathsf{c}_k\}$ be the set of all enabled rules after the call sequence $\mathbf{c}$. We claim that $\mathbf{c}.(\mathsf{c}_1.\mathsf{c}_2.\ldots.\mathsf{c}_k)^\omega$ is a rule-fair infinite computation of this protocol.

First, due to Epistemic Stuttering Lemma 21 for every $1 \leq j \leq k$ and $0 \leq i$ we have $\mathsf{EV}(\mathbf{c}.(\mathsf{c}_1.\mathsf{c}_2.\ldots.\mathsf{c}_k)^i.\mathsf{c}_1.\mathsf{c}_2.\ldots.\mathsf{c}_j) = \mathsf{EV}(\mathbf{c})$. This and Lemma 17 imply that all rules in R are enabled after any call sequence of the form $\mathbf{c}.(\mathsf{c}_1.\mathsf{c}_2.\ldots.\mathsf{c}_k)^i.\mathsf{c}_1.\mathsf{c}_2.\ldots.\mathsf{c}_j$ for any $j \in \{1, \ldots, k\}$ and $i \geq 0$. This shows that $\mathbf{c}.(\mathsf{c}_1.\mathsf{c}_2.\ldots.\mathsf{c}_k)^\omega$ is an infinite computation of this protocol. Also, we know that no other rule can be enabled after $\mathbf{c}.(\mathsf{c}_1.\mathsf{c}_2.\ldots.\mathsf{c}_k)^i.\mathsf{c}_1.\mathsf{c}_2.\ldots.\mathsf{c}_j$, because otherwise such a rule would already be enabled after $\mathbf{c}$ and so would belong to R. This shows that $\mathbf{c}.(\mathsf{c}_1.\mathsf{c}_2.\ldots.\mathsf{c}_k)^\omega$ is a rule-fair infinite computation of this protocol, because every rule enabled infinitely many times is executed infinitely many times.

Now, due to Lemma 23 there are only finitely many epistemically non-redundant call sequences to try as candidates for $\mathbf{c}$. For each such call sequence, by the Decidability of Semantics Theorem 13 it is decidable to determine whether it can be generated by the protocol.

For each such call sequence $\mathbf{c}$ we then check which rules, $\psi \to \mathsf{c}$, are enabled after $\mathbf{c}$. For each such a call c we subsequently compute $\mathsf{EV}(\mathbf{c})$ and $\mathsf{EV}(\mathbf{c}.\mathsf{c})$ using Lemma 16 and check whether they are all equal. By the above equivalence the considered gossip protocol does not rule-fairly terminate iff for some such call sequence $\mathbf{c}$ all just mentioned equalities hold. □

Finally, we show that agent-fair termination is decidable, as well.

Theorem 37 (Decidability of Agent-Fair Termination)**.** *Given a gossip protocol that uses as guards the formulas from $\mathcal{L}_{wn}$ it is decidable to determine whether it rule-fairly terminates.*

Proof. First we show that a gossip protocol does not agent-fairly terminate iff it can generate an epistemically non-redundant call sequence $\mathbf{c}$ such that each agent a enabled after $\mathbf{c}$ has an enabled rule $\psi \to \mathsf{c}$ such that $\mathsf{EV}(\mathbf{c}.\mathsf{c}) = \mathsf{EV}(\mathbf{c})$ holds. The reasoning is completely analogous to the one presented in the proof of the previous theorem, so we omit the details.

The rest of the proof is a small modification of the reasoning used in the above proof. As before there are only finitely many epistemically non-redundant call sequences $\mathbf{c}$ and for each such call sequence it is decidable to determine whether it can be generated by the protocol.

For each such call sequence $\mathbf{c}$ we then check which agents are enabled after $\mathbf{c}$. For each such agent we then check whether it has a rule $\psi \to \mathsf{c}$ that is enabled after $\mathbf{c}$ and such that $\mathsf{EV}(\mathbf{c}) = \mathsf{EV}(\mathbf{c}.\mathsf{c})$, where, again, this test is decidable by Lemma 16. By the initial equivalence the considered gossip protocol does not agent-fairly terminate iff for some such call sequence $\mathbf{c}$ it is possible to choose the rules in such a way that all the equalities hold. □

Lehmann, Pnueli, and Stavi (1981) considered in the context of nondeterministic programs a notion related to fairness, called ***justice*** (or ***weak fairness***). It can be readily introduced in the context of gossip protocols. An infinite computation is ***rule-just*** (resp. ***agent-just***) if all rules (resp. agents) that from a certain moment on are always enabled (in short,

eventually always enabled) are selected infinitely often. In the context of the nondeterministic programs the notions of infinite just and fair computations differ. However, this is not the case for the gossip protocols.

Indeed, it is straightforward to see that an infinite rule-fair computation is also rule-just. To show the converse consider an infinite rule-just computation $\overline{\mathsf{c}} = \mathsf{c}_1.\mathsf{c}_2.\ldots$ of a gossip protocol.

As in the proof of Theorem 36, on the account of Lemma 28 and the fact that there are only finitely many epistemic views, for some l we have $\mathsf{EV}(\mathsf{c}_1 \ldots \mathsf{c}_l) = \mathsf{EV}(\mathsf{c}_1 \ldots \mathsf{c}_l.\mathsf{c}_{l+1} \ldots \mathsf{c}_{l+i})$ for all $i > 0$.

Suppose now that a rule, say $\psi \to \mathsf{c}$, is infinitely often enabled. By Lemma 17 for all $i > 0$

$$(\mathcal{M}, \mathsf{c}_1.\ldots.\mathsf{c}_l) \models \psi \text{ iff } (\mathcal{M}, \mathsf{c}_1.\ldots.\mathsf{c}_l.\mathsf{c}_{l+1}.\ldots.\mathsf{c}_{l+i}) \models \psi,$$

so $\psi \to \mathsf{c}$ is eventually always enabled. Since $\overline{\mathsf{c}}$ is rule-just, this rule is selected infinitely often.

An analogous argument shows that infinite agent-just and agent-fair computations coincide. Consequently the notions of just computations do not lead to new notions of termination of gossip protocols.

11. Conclusions

In this paper we studied decidability questions concerning distributed epistemic gossip protocols considered by Apt et al. (2016). First we established that these protocols are implementable. We proved it by showing a more general statement, namely that the semantics of the introduced epistemic language $\mathcal{L}_{wn}$ is decidable. We also established that truth of the formulas of $\mathcal{L}_{wn}$ is decidable. We then used the developed apparatus to show that partial correctness of these gossip protocols is decidable, as well. Finally, we showed that the problems of determining termination and fair termination (in two different senses) of a gossip protocol are decidable, as well.

The above decidability results deal only with formulas without nested modalities. An interesting open question is whether they can be extended to formulas that admit nested modalities. The main stumbling block in generalizing our proofs is that, as Example 10 shows, the crucial Semantic Stuttering Lemma 9 cannot be extended to arbitrary formulas of $\mathcal{L}$.

These considerations lead to another interesting open problem. Gossip protocols studied by Apt et al. (2016) are parametric in the sense that they are formulated in such a way that they do not depend on the underlying graph (for instance a ring). The same is true in the case of the protocol discussed in Section 5. The results we proved here allow us only to consider each specific gossip protocol (for example for a ring formed by 15 agents or for a specific graph with 36 nodes) separately. What is needed is a decision procedure that would allow us to consider all instances of a protocol (for example for all rings or for all graphs) simultaneously. We conjecture that this decision problem is undecidable both for partial correctness and for termination.

Finally, it would be interesting to find a sound and complete axiomatization both of the logic $\mathcal{L}$ and of its extension that would allow one to carry out the correctness proofs of the gossip protocols axiomatically, in the style of Hoare's logic (1969).

In the above exposition starting from Section 6 we focused on the push-pull mode of communication. A completely analogous presentation can be given for the push and pull modes. In particular, all decidability results obtained in this paper also hold for the calls in the push mode or the pull mode. The approach is the same. It calls for a modification of the notions of a view and an epistemic view for, respectively, the push and the pull mode and for an appropriate modification of the proofs of the obtained results. The details are straightforward and omitted.

The semantics we introduced in Section 3 stipulates through the definition of $\mathbf{c}(\mathsf{s})$ that a call ab is not noted by any agent $c \notin \{a, b\}$. The same is the case for the calls $a \triangleright b$ and $a \triangleleft b$. Attamah et al. (2014b) studied different type of calls, namely

- ab^-, which stipulates that every agent $c \notin \{a, b\}$ noted that a called b,
- ab^0, which stipulates that every agent $c \notin \{a, b\}$ noted that some call took place, though not between whom,
- ab^+ which stipulates that every agent $c \notin \{a, b\}$ noted that possibly some call took place, though not between whom.

Each of these types of calls entails a notion of the equivalence relation on the call sequences that differs from the ones considered here for the calls ab, $a \triangleright b$ and $a \triangleleft b$. It would be interesting to check whether our decidability results hold for the above types of calls, as well. As a starting point for such considerations ? (?) recently provided a uniform semantics for these and other types of calls, including the ones considered in this paper.

Another issue interesting to study is a natural generalization of the considered setup to ***conference calls***. These are calls that involve two or more agents who exchange all the secrets.

As the overview of related work shows, there is no single setup for a study of gossip protocols. Kermarrec and van Steen (2007) identified three crucial aspects of gossip protocols: *peer selection*, *data exchange* and *data processing*. We studied here a specific instance of these three aspects. The peer selection is statically determined by the underlying graph topology, while data exchange is modelled by the introduced three modes of communication: push-pull, push and pull, each considered separately, with a simultaneous communication between more than one pair of agents disallowed. In turn, data processing is realized by exchanging, respectively passing, all available information. This selection, combined with the use of epistemic guards, led to a simple yet expressive framework in which partial correctness, termination and fair termination of the resulting gossip protocols are all decidable.

A different selection of the above three aspects changes the framework in a fundamental way. It would be interesting to see for which other realizations of the framework of Kermarrec and van Steen (2007) similar decidability results can be obtained.

The semantics considered here can be modified to tailor it to different initial assumptions. For example, in some cases correctness of a gossip protocol may depend on the initial knowledge of the underlying network of the agents (for example that it is a directed ring).

One can take this into account by assuming that each agent not only knows its own program, but also the structure of the underlying network. In other words, each agent knows which calls are syntactically correct. This leads, (see Apt et al., 2016, 2017), to a

modified definition of the gossip model in which instead of $\mathbf{C}$ one uses the set $\mathbf{C}_P$ of valid call sequences in a gossip protocol P.

Another natural approach would be to assume that each agent knows only his own network connections. This would lead to a different setup in which each agent i considers as valid a different set of call sequences $\mathbf{C}_i$ that he would use to evaluate formulas of the form $K_i\phi$. A final possibility, suggested as a future work by ? (?), would be to construct a semantics that takes into account that all agents know the gossip protocol they execute. It would be interesting to check which results remain valid for each of these three alternatives.

As a final interesting research topic we would like to mention the synthesis of distributed epistemic gossip protocols from epistemic specifications. For a related work on a synthesis of knowledge-based programs see, e.g., the work of van der Meyden and Wilke (2005).

Acknowledgments

The results established in this paper were originally reported, in a shortened form, by Apt and Wojtczak (2016) and Apt and Wojtczak (2017b). We thank the reviewers of these papers and three reviewers of the present paper for helpful comments. The first author was partially supported by the NCN grant nr 2014/13/B/ST6/01807. He would like to thank Davide Grossi and Wiebe van der Hoek for helpful discussions that led to this paper. The second author was partially supported by the EPSRC grants EP/M027287/1 and EP/P020909/1.

References

Apt, K. R., Grossi, D., & Van der Hoek, W. (2016). Epistemic protocols for distributed gossiping. In *Proc. of the 15th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 2015)*, Vol. 215 of *EPTCS*, pp. 51–66.

Apt, K. R., Kopczyński, E., & Wojtczak, D. (2017). On the computational complexity of gossip protocols. In *Proc. of the 26th International Joint Conference on Artificial Intelligence, (IJCAI 2017)*. To appear.

Apt, K. R., & Wojtczak, D. (2016). On decidability of a logic of gossips. In *Proc. of the 15th European Conference on Logics in Artificial Intelligence (JELIA 2016)*, Vol. 10021 of *Lecture Notes in Computer Science*, pp. 18–33. Springer.

Apt, K. R., & Wojtczak, D. (2017a). Common knowledge in a logic of gossips.. Submitted.

Apt, K. R., & Wojtczak, D. (2017b). Decidability of fair termination of gossip protocols. In *Proc. of LPAR 2017*. To appear.

Apt, K. R., Francez, N., & Katz, S. (1988). Appraising fairness in languages for distributed programming. *Distributed Computing*, *2*(4), 226–241.

Attamah, M., van Ditmarsch, H., Grossi, D., & Van der Hoek, W. (2014a). A framework for epistemic gossip protocols. In *Proc of the 12th European Conference on Multi-Agent Systems (EUMAS 2014)*, pp. 193–209.

Attamah, M., van Ditmarsch, H., Grossi, D., & Van der Hoek, W. (2014b). Knowledge and gossip. In *Proceedings of ECAI 2014*, Vol. 263 of *Frontiers in Artificial Intelligence and Applications*, pp. 21–26. IOS Press.

Cooper, M. C., Herzig, A., Maffre, F., Maris, F., & Régnier, P. (2016a). A simple account of multiagent epistemic planning. In *Proc. of ECAI 2016*, pp. 193–201. IOS Press.

Cooper, M. C., Herzig, A., Maffre, F., Maris, F., & Regnier, P. (2016b). Simple Epistemic Planning: Generalised Gossiping. In *Proc. of ECAI 2016*, Vol. 285 of *Frontiers in Artificial Intelligence and Applications*, pp. 1563–1564. IOS Press.

Dijkstra, E. W. (1975). Guarded commands, nondeterminacy and formal derivation of programs. *Communications of the ACM*, *18*, 453–457.

Fagin, R., Halpern, J. Y., Moses, Y., & Vardi, M. Y. (1997). Knowledge-based programs. *Distributed Computing*, *10*(4), 199–225.

Francez, N. (1986). *Fairness*. Springer-Verlag, New York.

Halpern, J. Y., & Zuck, L. D. (1992). A little knowledge goes a long way: Knowledge-based derivations and correctness proofs for a family of protocols. *Journal of the ACM*, *39*(3), 449–478.

Hedetniemi, S. M., Hedetniemi, S. T., & Liestman, A. L. (1988). A survey of gossiping and broadcasting in communication networks. *Networks*, *18*(4), 319–349.

Herzig, A., & Maffre, F. (2015). How to share knowledge by gossiping. In *Proc of the 13th European Conference on Multi-Agent Systems (EUMAS 2015), Revised Selected Papers*, Vol. 9571, pp. 249–263. Springer.

Herzig, A., & Maffre, F. (2017). How to share knowledge by gossiping. *AI Communications*, *30*(1), 1–17.

Hoare, C. A. R. (1969). An axiomatic basis for computer programming. *Communications of the ACM*, *12*, 576–580, 583.

Hoare, C. A. R. (1978). Communicating sequential processes. *Communications of the ACM*, *21*, 666–677.

Hromkovic, J., Klasing, R., Pelc, A., Ruzicka, P., & Unger, W. (2005). *Dissemination of Information in Communication Networks - Broadcasting, Gossiping, Leader Election, and Fault-Tolerance*. Texts in Theoretical Computer Science. An EATCS Series. Springer.

INMOS Limited (1984). *Occam Programming Manual*. Prentice-Hall International, Englewood Cliffs, N.J.

Kempe, D., Dobra, A., & Gehrke, J. (2003). Gossip-based computation of aggregate information. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pp. 482–491. IEEE.

Kermarrec, A., & van Steen, M. (2007). Gossiping in distributed systems. *Operating Systems Review*, *41*(5), 2–7.

König, D. (1927). Über eine schlußweise aus dem endlichen ins unendliche. *Acta Litt. Ac. Sci.*, *3*, 121–130.

Ladin, R., Liskov, B., Shrira, L., & Ghemawat, S. (1992). Providing high availability using lazy replication. *ACM Transactions on Computer Systems (TOCS)*, *10*(4), 360–391.

Lehmann, D. J., Pnueli, A., & Stavi, J. (1981). Impartiality, justice, and fairness: the ethics of concurrent termination. In Kariv, O., & Even, S. (Eds.), *Proceedings of International Colloquium on Automata Languages and Programming (ICALP '81)*, pp. 264–277, New York. Lecture Notes in Computer Science 115, Springer-Verlag.

Tijdeman, R. (1971). On a telephone problem. *Nieuw Archief voor Wiskunde*, *3(XIX)*, 188–192.

van der Meyden, R., & Wilke, T. (2005). Synthesis of distributed systems from knowledge-based specifications. In *Proceedings 16th International Conference CONCUR 2005*, Vol. 3653 of *Lecture Notes in Computer Science*, pp. 562–576. Springer.

van Ditmarsch, H., Grossi, D., Herzig, A., van der Hoek, W., & Kuijer, L. B. (2016). Parameters for epistemic gossip problems. In *Proc. of LOFT 2016*. `https://pdfs.semanticscholar.org/74b5/2c025f335ba487cac612019e39ce6c818448.pdf`.

van Ditmarsch, H., Kokkinis, I., & Stockmarr, A. (2017a). Reachability and expectation in gossiping. In An, B., Bazzan, A., Leite, J., Villata, S., & van der Torre, L. (Eds.), *PRIMA 2017: Principles and Practice of Multi-Agent Systems*, pp. 93–109, Cham. Springer International Publishing.

van Ditmarsch, H., van Eijck, J., Pardo, P., Ramezanian, R., & Schwarzentruber, F. (2017b). Epistemic protocols for dynamic gossip. *J. of Applied Logic*, *20*(C), 1–31.