Model Checking Social Network Models

Raúl Pardo Gerardo Schneider
Department of Computer Science and Engineering,
Chalmers | University of Gothenburg, Sweden.
pardo@chalmers.se gerardo@chalmers.se

A *social network service* is a platform to build social relations among people sharing similar interests and activities. The underlying structure of a social networks service is the *social graph*, where nodes represent users and the arcs represent the users' social links and other kind of connections. One important concern in social networks is *privacy*: what others are (not) allowed to *know* about us. The "logic of knowledge" (*epistemic logic*) is thus a good formalism to define, and reason about, privacy policies. In this paper we consider the problem of verifying knowledge properties over *social network models* (SNMs), that is social graphs enriched with *knowledge bases* containing the information that the users know. More concretely, our contributions are: i) We prove that the model checking problem for epistemic properties over SNMs is decidable; ii) We prove that a number of properties of knowledge that are sound w.r.t. Kripke models are also sound w.r.t. SNMs; iii) We give a satisfaction-preserving encoding of SNMs into *canonical* Kripke models, and we also characterise which Kripke models may be translated into SNMs; iv) We show that, for SNMs, the model checking problem is cheaper than the one based on standard Kripke models. Finally, we have developed a proof-of-concept implementation of the model-checking algorithm for SNMs.

1 Introduction

Social networks services (or simply *social networks*) are one of the most popular services on the Internet nowadays. One of the main concerns in social networks is that of privacy: most users are not in full control over what they share, and it is not uncommon that private and personal data is leaked to an unintended audience [8]. These concerns arise because users cannot determine (in a precise manner) who *knows* their personal information. One solution is to provide users with more fine grained control over who knows their information. Epistemic logic or "the logic of knowledge" [5] offers great precision and granularity for modelling and reasoning about the knowledge of the (users or *agents*) in a system.

In [11] we introduced $\mathscr{PPF}$, a formalism based on epistemic logic to specify privacy policies in social networks, and to enable a formal assessment on whether these policies are preserved. $\mathscr{PPF}$ consists of: i) A generic model for social networks (SNMs); ii) A knowledge-based logic ($\mathscr{KBL}$) to reason about the social network and privacy policies; iii) A formal language ($\mathscr{PPL}$) to describe privacy policies (based on $\mathscr{KBL}$). In [10], $\mathscr{PPF}$ was further extended by providing agents with a deductive engine to perform knowledge inferences, and including an operational semantics to model the dynamics of social networks.

$\mathscr{PPF}$ has been specifically designed for privacy policies for real social networks, and that is why the language $\mathscr{PPL}$ and the underlying logic $\mathscr{KBL}$ are interpreted over SNMs and not over Kripke models (*possible-worlds* semantics), which is the "standard" way to give semantics to epistemic logic. In Kripke models the uncertainty of the agents is modelled using an *accessibility relation*. This relation connects all the worlds in the model that an agent considers possible. If a formula is true in all of them, then the agent knows it. This does not correspond to the way users in real world social networks acquire and reason about information. Typically, when a user joins a social network, she knows none or a few

Submitted to:
GandALF 2017

facts about it. The system might suggest some friends that are retrieved from the user's phone contacts. As the user makes new friends and they share information, her knowledge starts to grow, and later from this set of accumulated knowledge users may derive new facts.

There are two main advantages in $\mathscr{PPF}$'s design (as opposed to standard Kripke models):

1. **It preserves the original structure of real social networks**. The models in $\mathscr{PPF}$ (SNMs) consist of the *social graph* [4] and a knowledge base per user. The topology of the social graph provides information regarding the relationships between users (e.g., friends, colleagues,...). The knowledge base gives semantics to the modality $K_i\varphi$ (user i knows φ). Knowledge bases are not a new invention, they are just an instance of the *syntactic* approach to modelling knowledge [7]. This structure is also important from the enforcement point of view since it facilitates the integration of the framework with the target social network.
2. **Checking whether a user knows something must be as efficient as possible**. The privacy policies that users can specify in $\mathscr{PPF}$ talk about knowledge, e.g., "Only my friends can know my location" or "Only my family can know that I am going to my father's birthday party". Therefore, the enforcement of $\mathscr{PPF}$ privacy policies mainly depends on how efficiently these checks are performed. Social networks have millions of users, who disclose tons of information per second. As a consequence, a slow enforcement mechanism would not work in practice. By splitting the users' knowledge in different knowledge bases, the complexity of checking whether a user knows a piece of information can be significantly reduced. In Section 6 we study the improvement in complexity of having separated knowledge bases as opposed to standard Kripke semantics.

The properties of knowledge related to human reasoning, present in Kripke models, have been studied for decades and they are well-understood [5]. On the other hand, the properties of knowledge in SNMs have not been throughly studied. Therefore, several questions need to be answered: i) What is the relation between SNMs and Kripke models? ii) Does this slightly different representation of knowledge preserve the same properties? iii) Is it possible to determine whether an epistemic formula written in $\mathscr{KBL}$ is satisfied on a given SNM?[1] In this paper we study in depth the answer to these questions providing evidence that $\mathscr{PPF}$ not only offers advantages from the practical point of view, but also models knowledge as traditionally understood and accepted in the epistemic logic literature.

More concretely, our contributions are: i) A proof that model checking $\mathscr{KBL}$ formulae over SNMs is decidable, the algorithm being an implementation of the satisfaction relation for $\mathscr{KBL}$ (Section 3); ii) A logical characterisation of a number of properties of knowledge for SNMs including *common* and *distributed knowledge* (Section 4). iii) A translation from SNMs into *canonical* Kripke models, together with a proof that satisfaction is preserved (Section 5); we also show that it is always possible to reconstruct the original SNM from the canonical Kripke model, by considering the state associated with the characteristic formulae (Section 5.1); iv) A formal comparison of the complexity of the model checking problem for SNMs and for Kripke models where we show that the former is more efficient (Section 6). Additionally, we provide a proof-of-concept implementation of the model-checking algorithm.[2] The extended version of this paper includes the proofs of all Theorems and Lemmas [12].

2 Preliminaries

Here we briefly recall First-Order Epistemic Logic [5], social network models and the logic $\mathscr{KBL}$ [10].

[1] Answering this question will also solve the model checking problem for privacy policies written in $\mathscr{PPL}$, as checking conformance of $\mathscr{PPL}$ is reduced to checking satisfaction of a $\mathscr{KBL}$ formula.

[2] `https://github.com/raulpardo/kbl-model-checker`

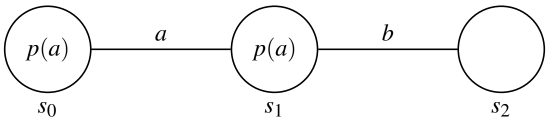

Figure 1: Relational Kripke structure

2.1 First-Order Epistemic Logic

We start with a set $\mathscr{T}$, consisting of *relation symbols* (p), *function symbols* (f) and *constants symbols* (c). Hereafter we will refer to $\mathscr{T}$ as the *vocabulary*. Each relation and function symbol has an implicit *arity* which corresponds to the number of arguments it takes. Function and relation symbols are interpreted over elements of a domain. We assume an infinite supply of variables, which we write as x, y and so on. We can form terms using constants, variables, and function symbols. Formally, a *term t* is recursively defined as follows: $t ::= c \mid x \mid f(\overrightarrow{t})$, where $\overrightarrow{t}$ represents a list of terms $t_1, \ldots, t_k$. An *atomic formula* is of the form $p(\overrightarrow{t})$ where p is a relation symbol. Let Ag be a set of *agents*, $i \in Ag$ and $G \subseteq Ag$, the syntax of *First-Order Epistemic Logic* (FOEL), denoted as $\mathscr{L}$, is recursively defined as follows [5]:

$$\varphi ::= p(\overrightarrow{t}) \mid \varphi \wedge \varphi \mid \neg\varphi \mid \forall x.\varphi \mid K_i\varphi$$

The remaining epistemic modalities are defined as $S_G\varphi \triangleq \bigvee_{i\in G} K_i\varphi$ and $E_G\varphi \triangleq \bigwedge_{i\in G}\varphi$. The intuitive meaning of the modalities is the following: $K_i\varphi$, agent i knows φ; $E_G\varphi$, everyone in the group G knows φ; $S_G\varphi$, someone in the group G knows φ. The semantics of FOEL formulae is given using *relational Kripke models*. In what follows we sometimes omit relational and write Kripke models.

Definition 1 ([5]). *A* relational Kripke Model *is a tuple of the form* $\langle S, \pi, \{\mathscr{K}_i\}_{i\in Ag}\rangle$*, where:*

- *S is a non-empty set of* states *(or* worlds*).*
- *$\pi : S \to \mathscr{A}$ is a function that associates to each world a* relation structure *for a fixed vocabulary $\mathscr{T}$. As usual, $\mathscr{A}$ consists of a domain $dom(\mathscr{A})$, an assignment of a k-ary relation $P^{\mathscr{A}} \subseteq dom(\mathscr{A})^k$ for each relation symbol, an assignment of a k-ary function $f^{\mathscr{A}} : dom(\mathscr{A})^k \to dom(\mathscr{A})$ for each function symbol and an assignment of a member $c^{\mathscr{A}}$ of the domain for each constant symbol.*
- *$\{\mathscr{K}_i\}_{i\in Ag}$ where $\mathscr{K}_i \subseteq S \times S$ is an* accessibility relation *between states.*

Example 1. *Let us consider a Kripke structure consisting of agents a and b, states s_0, s_1 and s_2, a predicate p with arity 1 and relations $\mathscr{K}_a = \{(s_0,s_1),(s_1,s_0)\}$ and $\mathscr{K}_b = \{(s_1,s_2),(s_2,s_1)\}$. We assume here that all relational structures $\pi(s_n)$ have a common domain $dom(\mathscr{A}) = \{a,b\}$, i.e., Ag. Moreover, $a \in P^{\pi(s_0)}$ and $a \in P^{\pi(s_1)}$. Fig. 1 shows a graphical representation of the described model.* □

Usually free variables and terms are interpreted using a *valuation* function, which is parametrised with a relational structure depending of the state of the Kripke model in which the formula is evaluated. For simplicity, in this paper we will assume that formulae in $\mathscr{L}$ do not contain free variables (i.e., all variables are quantified) and the interpretation of functions and constants is the same independently of the state where they are evaluated. Thus, we assume that terms are implicitly interpreted and we do not include the valuation function as a parameter in the satisfaction relation below.

Definition 2 ([5]). *Given a non-empty set of agents Ag, a relational Kripke model M, a state $s \in M$, agents $i, j, u \in Ag$ and a finite set of agents $G \subseteq Ag$, we define what it means for $\varphi \in \mathscr{L}$ to be* satisfied *by (M,s), written $(M,s) \vDash \varphi$, as shown in Table 1.*

We say that a formula φ is *valid in a Kripke model M*, and we write $M \vDash \varphi$, if $\forall s \in M$ $(M,s) \vDash \varphi$. Moreover, we say that φ is *valid*, denoted as $\vDash \varphi$, if for all Kripke models M it holds $M \vDash \varphi$.

$(M,s) \vDash p(t_1,\dots,t_k)$	iff	$(t_1,\dots,t_k) \in P^{\pi(s)}$
$(M,s) \vDash \neg\varphi$	iff	$(M,s) \not\vDash \varphi$
$(M,s) \vDash \varphi_1 \wedge \varphi_2$	iff	$(M,s) \vDash \varphi_1$ and $(M,s) \vDash \varphi_2$
$(M,s) \vDash \forall x.\varphi$	iff	for all $v \in dom(\pi(s))$, $(M,s) \vDash \varphi[v/x]$
$(M,s) \vDash K_i\varphi$	iff	$(M,t) \vDash \varphi$ for all t such that $(s,t) \in \mathscr{K}_i$

Table 1: Satisfaction relation over Kripke models

Example 2. *Let M be the model presented in Fig. 1. It holds that $(M,s_0) \vDash K_a p(a)$, since $p(a)$ holds in s_0 and in all the states accessible for a from s_0 (only s_1). It also holds that $(M,s_1) \vDash \neg K_b p(a)$, since in one of the states that b considers possible $p(a)$ is not true. In particular, $(M,s_2) \vDash \neg p(a)$.*

2.2 $\mathscr{KBL}$ and Social Network Models

$\mathscr{KBL}$ is a *knowledge-based logic* for social networks. It contains all the knowledge modalities presented in $\mathscr{L}$, and additionally, it includes two special types of predicates. The connection and action predicates. Connection predicates represent the "social" connections between users. For instance, friends, colleagues, family, co-workers, and so forth. Action predicates model the permitted actions a user may execute. For example, Alice can send a friend request to Bob or Alice can join events created by Bob. Note that action predicates are not deontic modalities. Hereafter we use $\mathscr{C}$ and Σ to denote sets of indexes for connections and permissions, respectively. As before the set Ag represents a set of agents in the system.

Definition 3. *Given $i, j \in Ag$, a set of predicate symbols $\mathscr{P}$ such that $a_n(i,j), c_m(i,j), p(\vec{t}) \in \mathscr{P}$ where $m \in \mathscr{C}$ and $n \in \Sigma$, and $G \subseteq Ag$, the syntax of the* knowledge-based logic *$\mathscr{KBL}$ is inductively defined as:*

$$\varphi \quad ::= \quad c_m(i,j) \mid a_n(i,j) \mid p(\vec{t}) \mid \varphi \wedge \varphi \mid \neg\varphi \mid \forall x.\varphi \mid K_i\varphi$$

As before, the remaining epistemic modalities are defined as $S_G\varphi \triangleq \bigvee_{i\in G} K_i\varphi$ and $E_G\varphi \triangleq \bigwedge_{i\in G} \varphi$.

Terms and atomic formulae are defined as for $\mathscr{L}$. $\mathscr{F}_{\mathscr{KBL}}$ denotes the set of *well-formed formulae* of $\mathscr{KBL}$ (category φ of Def. 3).

Social networks are usually modelled as graphs where nodes represent the users (or agents), and edges represent different relationships among agents or any other social network specific information [4]. These graphs are known as *social graphs*. Here we enrich social graphs with information about the agents knowledge, permissions, connections and privacy policies as defined below.

Definition 4 (Social Network Model)**.** *Given a set of $\mathscr{KBL}$ formulae $\mathscr{F}$, a set of privacy policies Π, and a finite set of agents $Ag \subseteq \mathscr{AU}$ from a universe $\mathscr{AU}$, a* social network model (SNM) *is a social graph of the form $\langle Ag, \mathscr{A}, KB, \pi\rangle$, where*

- *Ag is a nonempty finite set of* nodes *representing the agents of the social network.*
- *$\mathscr{A}$ is a first-order relational structure for the fixed vocabulary of the* SNM*, which as before, consists of a finite domain $dom(\mathscr{A})$[3], an assignment of a k-ary relation $P^{\mathscr{A}} \subseteq dom(\mathscr{A})^{\mathscr{A}}$ for each predicate symbol, an assignment of a k-ary $f^{\mathscr{A}} : dom(\mathscr{A})^k \rightarrow dom(\mathscr{A})$ for each function symbol and assignment of a member $c^{\mathscr{A}}$ of the domain for each constant symbol.*

[3] For the sake of clarity in definitions and proofs and w.l.o.g. we have only considered a single finite domain in the formal definition. However, in the rest of the paper we will assume that we have a finite set of finite domains. For instance, we can have $dom(\mathscr{A})$ consisting of the domain of agents, timestamps, indexes for pictures, etc. All the results also hold in SNMs consisting of multiple domains as we consider a finite number of finite domains.

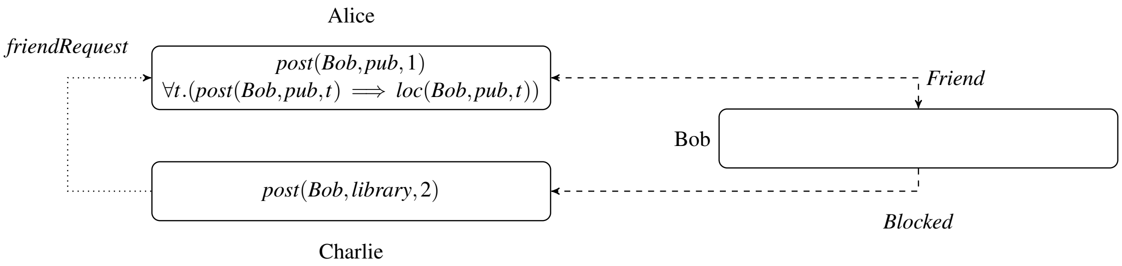

Figure 2: Example of Social Network Model

- *$KB : Ag \to 2^{\mathscr{F}}$ is a function that returns a finite set of accumulated knowledge for each agent, stored in what we call the* knowledge base *of the agent. We write KB_i to denote $KB(i)$.*
- *$\pi : Ag \to 2^{\Pi}$ is a function that returns a finite set of privacy policies for each agent. We write π_i to denote $\pi(i)$.*

The shape of the relational structure $\mathscr{A}$ depends on the concrete the social network. Connections and permission actions between agents, i.e., edges of the social graph, are represented as families of binary relations, $\{C_i\}_{i\in\mathscr{C}} \subseteq 2^{Ag\times Ag}$ and $\{A_i\}_{i\in\Sigma} \subseteq 2^{Ag\times Ag}$ over the domain of agents. Sometimes, we write an atomic formula, e.g. $friends(a,b)$ to denote that the elements $a, b \in Ag$ belong to a binary relation, $friends$, defined over pairs of agents as expected. $\mathscr{SN}$ denotes the universe of all possible SNMs.

The knowledge base KB_i of each agent i contains the explicit knowledge that the agent has. Besides this explicit knowledge, agents also know anything that can be derived from formulae in their knowledge bases (using the **KD4** axiomatisation of epistemic logic [5]).

Definition 5. *A* derivation *of a formula $\varphi \in \mathscr{F}_{\mathscr{KBL}}$, is a finite sequence of formulae $\varphi_1, \dots, \varphi_n = \varphi$ where each φ_i, for $1 \leq i \leq n$, is either an instance of the axioms or the conclusion of one of the derivation rules of the **KD4** axiomatisation which premises have already been derived, i.e., it appears as φ_j with $j < i$.*

Given a set of formulae $\Gamma \in 2^{\mathscr{F}_{\mathscr{KBL}}}$, we write $\Gamma \vdash \varphi$ to denote that φ can be derived from Γ.

Additionally, we impose two assumptions in users' knowledge bases:

i) φ and $\neg\varphi$ cannot be derivable in the same KB_i. It prevents users from having inconsistent knowledge.

ii) If φ is in i's knowledge base, $K_i\varphi$ is also there. In this way we make users aware of their knowledge.

These assumptions are formalised as the following properties:

Definition 6 (Knowledge Consistency)**.** *For all $i \in Ag$ and formulae $\varphi \in \mathscr{F}_{\mathscr{KBL}}$, if $KB_i \vdash \varphi$ then $KB_i \not\vdash \neg\varphi$.*Enforcing knowledge consistency is straightforward. Before adding any formula φ to KB_i we check that $KB_i \cup \{\varphi\} \not\vdash \neg\varphi$.

Definition 7 (Self-Awareness)**.** *For all $i \in Ag$ and formulae $\varphi \in \mathscr{F}_{\mathscr{KBL}}$, if $KB_i \vdash \varphi$ then $KB_i \vdash K_i\varphi$.*

Remark 1. Self-awareness is not equivalent to the necessitation rule in **KD4**. Necessitation states that if a φ is provable from no assumptions then $K_i\varphi$ is provable from no assumptions as well [5]. That is, $\frac{\vDash \varphi}{\vDash K_i\varphi}$. It requires φ to be a tautology. On the other hand, self-awareness states that if φ is derivable from i's knowledge, then $K_i\varphi$ is also derivable. For example, $\varphi \vee \neg\varphi$ is provable from no assumptions. Therefore, from axiom A1 it is derivable $KB_i \vdash \varphi \vee \neg\varphi$ for all KB_i. Consequently, by necessitation it also holds that $KB_i \vdash K_j\varphi \vee \neg\varphi$ for all KB_i and $j \in Ag$. However, consider now a predicate $p(\overrightarrow{t})$ which is not

$SN \vDash p(\vec{t})$	iff	$p(\vec{t}) \in KB_e$
$SN \vDash c_m(i,j)$	iff	$(i,j) \in C_m$
$SN \vDash a_n(i,j)$	iff	$(i,j) \in A_n$
$SN \vDash \neg\varphi$	iff	$SN \nvDash \varphi$
$SN \vDash \varphi \wedge \psi$	iff	$SN \vDash \varphi$ and $SN \vDash \psi$
$SN \vDash \forall x.\varphi$	iff	for all $v \in dom(\mathscr{A})$, $SN \vDash \varphi[v/x]$
$SN \vDash K_i\varphi$	iff	$KB_i \vdash \varphi$

Table 2: $\mathscr{KBL}$ satisfaction relation

derivable from no assumptions. It does not hold that $KB_i \vdash p(\vec{t})$ for all KB_i. There is no axiom which includes $p(\vec{t})$ in the set of derivations of $\vdash$. Nevertheless, self-awareness says that if $KB_i \vdash p(\vec{t})$ then $KB_i \vdash K_i p(\vec{t})$. Note that, unlikely necessitation, we use the same agent i in KB_i and $K_i p(\vec{t})$.

Example 3. *Let SN be an SNM consisting of three agents Alice, Bob and Charlie, $Ag = \{Alice, Bob, Charlie\}$; the friend request action, $\Sigma = \{friendRequest\}$; and the connections Friend and Blocked, $\mathscr{C} = \{Friend, Blocked\}$. Here, we define $dom(\mathscr{A})$ to be a finite set of timestamps.*

Fig. 2 shows a graphical representation of SN. In this model the dashed arrows represent connections. Note that the Friend connection is bidirectional, i.e., Alice is friend with Bob and vice versa. On the other hand, it is also possible to represent unidirectional connections, as Blocked; in SN Bob has blocked Charlie. Permissions are represented using a dotted arrow. In this example, Charlie is able to send a friend request to Alice.

The predicates inside each node represent the agents' knowledge, e.g., Alice has $post(Bob, pub, 1)$ in her knowledge base, meaning that she knows that Bob posted at time 1 that he was in a pub. Similarly, Charlie's knowledge base contains the predicate $post(Bob, library, 2)$ meaning that at time 2 Bob posted that he was in the library. Agents' nodes can also contain more complex $\mathscr{KBL}$ formulae that may increase their knowledge. For instance, Alice knows $loc(Bob, pub, 1)$ implicitly. Alice can in fact derive it by Modus Ponens, *from $post(Alice, pub, 1)$ and $\forall t.(post(Alice, pub, t) \implies loc(Bob, pub, t))$. The variable t ranges over $dom(\mathscr{A})$, which, as mentioned earlier, consists in a finite set of timestamps. Being able to derive $loc(Bob, pub, 1)$ means that Alice knows that Bob's location at time 1 was a pub.*

The satisfaction relation for $\mathscr{KBL}$ formulae, interpreted over SNMs, is defined as follows.

Definition 8. *Given an SNM $SN = \langle Ag, \mathscr{A}, KB, \pi \rangle$, agents $i, j \in Ag$, formulae $\varphi, \psi \in \mathscr{F}_{\mathscr{KBL}}$, a finite set of agents $G \subseteq Ag$, $m \in \mathscr{C}$ and $n \in \Sigma$, the* satisfaction relation *$\vDash \subseteq \mathscr{SN} \times \mathscr{KBL}$ is defined in Table 2.*

The intuition behind the semantic definition of the knowledge modality is different in $\mathscr{KBL}$ from that of epistemic logic. As shown in Table 1, the accessibility relation in Kripke models captures the *uncertainty* of the agents. It models all the states that an agent consider possible and knowledge is acquired when a given formula is true in all those states. In SNMs, knowledge is explicitly present in the knowledge bases of the agents, hence modelling what the agents know rather than what they consider possible. A given formula is known by an agent if it is present in her knowledge base or if she can derive it from her knowledge. We use a special agent called *environment* (or simply e) which defines the truth of atomic formulae of the type $p(\vec{t})$. The environment's knowledge base (KB_e) contains all predicates which are true in the real world. For instance, $loc(Alice, Sweden)$ is in KB_e only if Alice's location is Sweden or, similarly, only if Bob's age is 20 the predicate $age(Bob, 20)$ is in KB_e.

Example 4. *Let SN be the SNM in Fig. 2. As described in Example 3, Alice knows that Bob posted that he was in a pub at time 1, meaning that $SN \vDash K_{Alice} post(Bob, pub, 1)$ holds. Indeed, it holds since*

$post(Bob,pub,1)$ is in the knowledge base of Alice, i.e., $post(Bob,pub,1) \in KB_{Alice}$ and therefore it can be derived $KB_{Alice} \vdash post(Bob,pub,1)$ (1). Though not explicitly stated, it is possible for Alice to derive that Bob's location at time 1 was a pub, meaning that $SN \models K_{Alice} loc(Bob,pub,1)$ (2) should hold. Following the semantics of K_i in Table 2, the previous formula is true iff $KB_{Alice} \vdash loc(Bob,pub,1)$. Fig. 2 shows that KB_{Alice} contains the formula $\forall t.(post(Bob,pub,t) \implies loc(Bob,pub,t))$ (3)—where t is a timestamp —therefore the deductive engine derives $post(Bob,pub,1) \implies loc(Bob,pub,1)$ (4). From (1) and (4), by modus ponens *we can derive $loc(Bob,pub,1)$, i.e., $KB_{Alice} \vdash loc(Bob,pub,1)$, hence (2) holds.*

3 Model checking SNMs

In this section we present a model checking algorithm that directly implements the semantics of $\mathscr{KBL}$ in Table 2, and we show that model checking is decidable under the following assumptions:

Assumption 1. *All domains are finite.*

Assumption 2. *All functions are computable.*

These assumptions are present in all real social networks. Domains in SNMs might be, the set of users, posts, pictures, likes, tags and so on. In practice at any moment in time there is a finite amount of any of these elements. Consequently, when having a universal quantification over a domain it is reasonable to consider only the finite set of elements in the domain at that concrete moment in time. Furthermore, we assume that functions in $\mathscr{KBL}$ terms must be computable. As mentioned in the introduction, $\mathscr{KBL}$ is a logic embedded in a framework to express privacy policies. The framework includes the notion of instantiation where all the elements of SNMs are instantiated for a concrete social network. For instance, in [11] we presented the instantiations of Facebook and Twitter. In these instantiations functions were used to retreive information, e.g., $followers(u)$ which returns all the followers of the user or $friends(u)$ which returns all the friends of u. Another type of functions could be $weather(London)$ or $location(u)$, which return the current weather in London and u's current location, respectively. Therefore, computable functions are enough for the practical use of the logic.

Theorem 1. *Let SN be an SNM and $\varphi \in \mathscr{F}_{\mathscr{KBL}}$ be a formula. Determining whether $SN \models \varphi$ is decidable.*

Proof. We show decidability of the model checking problem for $\mathscr{KBL}$ by presenting an algorithm which implements the semantics of Table 2,

First, we expand the universal quantifiers in φ by inductively transforming each subformula $\forall x.\varphi'$ into a conjunction with one conjunct $\varphi'[v/x]$ for each element v of the domain $dom(\mathscr{A})$. Given that the domain is finite (see Assumption 1), it always terminates and results in a quantifier free formula. Secondly, we compute all functions and replace all constants with an element of the domain according to the assignment in $\mathscr{A}$. From Assumption 2, we can deduce that this step always terminates. After this step we are left with a quantifier free formula without functions or constant symbols. Finally, we inductively show that all the elements of the formula (see Def. 3) can be computed.

- Checking $c_m(i,j)$ and $a_n(i,j)$ can be performed in constant time, simply by checking $(i,j) \in C_m$ or $(i,j) \in A_n$, respectively.
- Checking $p(\overrightarrow{t})$ requires the query $p(\overrightarrow{t}) \in KB_e$ to the environment's knowledge base. It can be performed in constant time.
- $\neg\varphi$ and $\varphi_1 \wedge \varphi_2$ can be done in constant time, using the induction hypothesis.
- $K_i\varphi$ requires a query to the epistemic engine to determine $KB_i \vdash \varphi$. Solving the previous query is a decidable problem [5].

The algorithm goes recursively from the top most element of φ to the bottom. □

In Section 6 we study the complexity of this algorithm and compare it to that of model checking in traditional Kripke models. Nevertheless, in order to provide a fair comparison, we first show that the same set of properties of knowledge that are sound w.r.t. Kripke models are also sound w.r.t. SNMs.

4 Properties of Knowledge in SNMs

Here we explore properties of knowledge in SNMs. In particular, we consider the axioms of some of the standard axiomatisations for epistemic logic, and prove that such axioms are sound with respect to SNMs.

In [5] Fagin *et al.* show which properties of knowledge are sound w.r.t. Kripke models depending on the type of accessibility relation of the model. For instance, the following axiom is sound w.r.t. the set of Kripke models where the accessibility relation is reflexive: (A3) $K_i\varphi \implies \varphi$.

These properties of knowledge comprise the different axiomatisations of epistemic logic. In SNMs the properties of knowledge will depend on the axiomatisation from epistemic logic [5] that we choose for $\vdash$. As we described in Def. 5, $\vdash$ includes all the axioms and derivation rules from **KD4**.

In epistemic logic one can talk about *knowledge* or *belief* depending on the properties (or axiomatisations) that are sound w.r.t. a particular set of Kripke models. Axiom A3 is commonly called *Knowledge axiom*. It means that the facts agents know are true. When this axiom is not present, the "knowledge" of the agents is regarded as belief. As you might have noticed, in SNMs the truth of the facts that the agents know is not linked to whether they are true or not. For example, imagine that Alice knows that Bob and Charlie are friends, i.e., $K_{Alice}friend(Bob,Charlie)$, which is true iff $KB_{Alice} \vdash friend(Bob,Charlie)$. This is not connected to the actual truth of the predicate $friend(Bob,Charlie)$, which holds iff $(Bob,Charlie) \in C_{Friend}$. When the knowledge axiom is not present, some philosophers argue that it is required that the beliefs of the agents are consistent. This is captured by the following axiom, where $\perp$ represents *falsum*: (D) $\neg K_i\perp$.

In Kripke models, axiom D is present when the accessibility relation is serial [5]. In SNMs, we assume agents' knowledge bases to be consistent (see Def. 6). Therefore, $\perp$ cannot be derived.

Lemma 1. *Axiom D is sound with respect to SNMs.*

As we mentioned in the introduction, $\mathscr{KBL}$ and SNMs were developed in the context of a privacy policy framework for social networks [11, 10]. In privacy policies it is more natural to write "Alice cannot know my location" than "Alice cannot belief my location". Because of this, we chose to talk about knowledge, even though we are dealing with an axiomatisation for belief.

The most basic set of properties for Kripke models, i.e., the set of properties that are sound w.r.t. Kripke models with no conditions in their accessibility relation, is the **K** axiomatisation [5]. It consists of two axioms and two inference rules. Given $\varphi \in \mathscr{L}$ and $i \in Ag$,

A1. All (instances of) first-order tautologies,
A2. $(K_i\varphi \wedge K_i(\varphi \implies \psi)) \implies K_i\psi$,
R1. From φ and $\varphi \implies \psi$ infer ψ,
R2. From φ infer $K_i\varphi$ where φ must be provable from no assumptions.

Lemma 2. ***K** is sound with respect to SNMs.*

The axioms and inferences rules of **K**, together with axiom D comprises the axiom system **KD**. Nevertheless, there exist two more axioms that are normally present in knowledge and belief axiomatisations, the so called *positive introspection* (A4) and *negative introspection* (A5) [5]. The former expresses that agents in the system are aware of their knowledge, the latter means that agents know everything that they do not know. Given $\varphi \in \mathscr{L}$ and $i \in Ag$

A4. $K_i\varphi \implies K_iK_i\varphi$,
A5. $\neg K_i\varphi \implies K_i\neg K_i\varphi$.

Lemma 3. *Axiom A4 is sound with respect to SNMs.*

Lemma 4. *Axiom A5 is not sound with respect to SNMs.*

A4 follows from our assumption that agents are self-aware of their knowledge (see Def.7). On the other hand, A5 does not follow given the current set of assumptions in knowledge bases. An agent's knowledge base does not contain any knowledge regarding what she does not know, unless it is explicitly inserted.

The axiomatisation **K** together with axioms D and A4 forms the so-called **KD4** axiomatisation. We thus have the following result for SNMs.

Theorem 2. ***KD4*** *is sound with respect to SNMs.*

Common Knowledge

Here we introduce the notion of *common knowledge*, which we represent using the modality C_G where G is a group of agents. A fact becomes common knowledge when everybody knows it, and also, everyone knows that everyone knows it, and so forth. This is a useful concept in the social network setting. Consider the effect of publishing a post $p(\overrightarrow{t})$ in a social network. After posting, the owner of the post and the audience will know the post, $E_{\{owner\}\cup Audience}\ p(\overrightarrow{t})$. Moreover, the owner also will know that everyone who was included in the audience will know the post, $K_{owner}E_{Audience}\ p(\overrightarrow{t})$. But even more, each of the users in the audience will know that each other knows the post, i.e. $E_{\{owner\}\cup Audience}E_{\{owner\}\cup Audience}\ p(\overrightarrow{t})$ and so on. The traditional definition of common knowledge [5] over Kripke models accurately captures the described effect. Given a Kripke model M, a state $s \in M$, a formula $\varphi \in \mathscr{L}$ and a set of agents G, common knowledge is defined as follows: $(M,s) \vDash C_G\varphi$ iff $(M,s) \vDash E_G^k\varphi$ for $k = 1\ldots$ where $E_G^0\varphi = \varphi$ and $E_G^{k+1}\varphi = E_G\varphi E_G^k\varphi$. The definition of common knowledge for SNMs is analogous to the one above.

Definition 9. *Given an SNM SN, a formula $\varphi \in \mathscr{F}_{\mathscr{K}\mathscr{B}\mathscr{L}}$ and a set of agents G, common knowledge is defined as follows: $SN \vDash C_G\varphi$ iff $SN \vDash E_G^k\varphi$ for $k = 1\ldots$*

Given formulae $\varphi, \psi \in \mathscr{L}$, the set $G \subseteq Ag$ and $i \in Ag$, the following axiomatisation characterises common knowledge [5]:
C1. $E_G\varphi \iff \bigwedge_{i\in G} K_i\varphi$,
C2. $C_G\varphi \iff E_G(\varphi \wedge C_G\varphi)$,
RC1. From $\varphi \implies E_G(\psi \wedge \varphi)$ infer $\varphi \implies C_G\psi$ where $\varphi \implies E_G(\psi \wedge \varphi)$ must be provable from no assumptions.

Lemma 5. *The axioms C1 and C2, and the rule RC1 are sound w.r.t. SNMs.*

Distributed Knowledge

In this section we introduce the distributed knowledge operator, represented by the modality D_G. A fact becomes distributed knowledge in the group of agents G when it is known by combining the knowledge of all individual agents. It can be seen as a wise agent. In Kripke models, distributed knowledge is defined by removing possible states, i.e., removing uncertainty. Formally, $(M,s) \vDash D_G\varphi$ iff $(M,t) \vDash \varphi$ for all t such that $(s,t) \in \bigcap_{i\in G}\mathscr{K}_i$. We define distributed knowledge as the union of all the explicit knowledge that all the agents in G have and everything that can be derived from it.

Definition 10 (Distributed knowledge)**.** *Given an SNM SN, a formula $\varphi \in \mathscr{F}_{\mathscr{K}\mathscr{B}\mathscr{L}}$ and a set of agents G, distributed knowledge is defined as follows: $SN \vDash D_G\varphi$ iff $\bigcup_{i\in G} KB_i \vdash \varphi$.*

The following axioms characterise distributed knowledge [5]:
D1. $D_{\{i\}}\varphi \iff K_i\varphi,\ i = 1,\dots,n$,
D2. $D_G\varphi \implies D_{G'}(\varphi)$ if $G \subseteq G'$,
DA2 and DA4. Axioms A2 and A4 of **KD4**, K_i with D_G in each axiom.

Note that axiom D is not required because we work with a belief axiomatisation [5]. Therefore, it is possible for a group of agents to have inconsistent distributed beliefs. In what follows, we show that this axiomatisation for Kripke models is sound with respect to SNMs as well.

Lemma 6. *Axioms D1 and D2, together with the axioms A2 and A4 of the* ***KD4****-axiomatisation (replacing the modality K_i with the modality D_G) are sound w.r.t. SNMs.*

5 Translation of SNMs into Kripke Models

In this section, we show that SNMs can be encoded into Kripke models. Our proof is constructive, starting from an SNM we give a procedure to build a *canonical* Kripke model, and we prove that satisfaction is preserved when interpreting $\mathscr{KBL}$ formulae as epistemic logic formulae.

For epistemic logic, Fagin *et al.* show that it is possible to construct a canonical Kripke model which satisfies a given formula φ [5], provided that φ is consistent with respect to some of the axiomatisations of knowledge. A formula φ is ***KD4****-consistent* if $\neg\varphi$ cannot be derived. A set of formulae is **KD4**-consistent if the conjunction of all the formulae in the set is **KD4**-consistent. We say that a set of formulae Φ is *maximal* ***KD4****-consistent* with respect to the language $\mathscr{L}$, if Φ is **KD4**-consistent and for all φ in $\mathscr{L}$ but not in Φ, the set $\Phi \cup \{\varphi\}$ is not **KD4**-consistent. In what follows, we describe the procedure of how to construct a canonical Kripke model for a **KD4**-consistent formula. We will follow a similar approach when translating SNMs into Kripke models.

Definition 11 (Canonical Kripke model for **KD4**[5])**.** *Consider a* **KD4**-consistent *formula φ. Let $Sub(\varphi)$ be the set of all subformulae of φ. We define $Sub^+(\varphi)$ to be the set of all subformulae and their negations, i.e. $Sub^+(\varphi) = Sub(\varphi) \cup \{\neg\psi \mid \psi \in Sub(\varphi)\}$. We also define $Con(\varphi)$ to be the set of maximal* ***KD4****-consistent subsets of $Sub^+(\varphi)$. Given a set of formulae $\Theta \subseteq \mathscr{L}$, we define $\Theta/K_i = \{\varphi \mid K_i\varphi \in \Theta\}$. The canonical Kripke model for φ is defined as follows: $M_\varphi = \langle S_\varphi, \pi, \{\mathscr{K}_i\}_{i\in Ag}\rangle$ where $S_\varphi = \{s_\Theta \mid \Theta \in Con(\varphi)\}$, $\mathscr{K}_i = \{(s_\Theta, s_\Psi) \mid \Theta/K_i \subseteq \Psi/K_i,\ \Theta/K_i \subseteq \Psi\}$ and*

$$\pi(s_\Theta)(p(t_1,\dots,t_k)) = \begin{cases} \textbf{\textit{true}} & \textit{if } p(t_1,\dots,t_k) \in \Theta \\ \textbf{\textit{false}} & \textit{if } p(t_1,\dots,t_k) \notin \Theta \end{cases}$$

Fagin *et al.* show that φ is satisfiable in the resulting canonical Kripke model [5, Theorem 3.2.4]. The set of Kripke models that are sound and complete with respect to **KD4** are the ones with a serial and transitive accessibility relation. The accessibility relation of the previous canonical Kripke model is, as shown in [5, Theorem 3.2.4], serial and transitive. We denote the set of Kripke models with the previous type of accessibility relation as $\mathscr{M}^{lt}$.

The canonical Kripke model will have at most $2^{|\varphi|}$ states, as shown in [5, Theorem 3.2.4] where $|\varphi|$ is the length of the formula φ. Even though it is finite, this approach of constructing a Kripke model can lead to an exponential growth of the size of the model. For example, if we assume that the knowledge of the agents increases monotonically, i.e., agents do not forget any knowledge they have previously obtained, then the size of φ will have a lower bound, from which its size will only grow, and consequently, the size of the corresponding canonical Kripke model. In what follows, we define a function which takes an SNM and converts it into the corresponding canonical Kripke model.

First we describe how to construct a set containing all the true formulae in an SNM, called the characteristic set of the social network.

Definition 12. *The* characteristic set *of an SNM SN, denoted as Φ_{SN}, is constructed as follows:* $\Phi_{SN} = \{p(\overrightarrow{t}) \mid p(\overrightarrow{t}) \in KB_e\} \cup \{K_i\varphi \mid \varphi \in KB_i\} \cup \{c(i,j) \mid (i,j) \in C_c, c \in \mathscr{C}\} \cup \{a(i,j) \mid (i,j) \in A_a, a \in \Sigma\}$.

Moreover, we define the *characteristic formula* of an SNM.

Definition 13. *Given a characteristic set, Φ_{SN}, of an SNM SN, its* characteristic formula*, denoted as φ_{SN}, is defined as $\varphi_{SN} = \bigwedge_{\psi \in \Phi_{SN}} \psi$.*

We will use the characteristic formula of an SNM to create the corresponding Kripke model, therefore we must show that this formula is **KD4**-consistent.

Lemma 7. *For all $SN \in \mathscr{SN}$, φ_{SN} is **KD4**-consistent.*

We are now ready to provide our translation from SNMs into canonical Kripke models.

Definition 14 (Kripke transformation function)**.** *Let $\mathscr{KT} : \mathscr{SN} \rightarrow \mathscr{M}^{lt}$ be a function which takes an SNM and converts it to the corresponding Kripke model as follows. Given an $SN \in \mathscr{SN}$, $\mathscr{KT}(SN)$ is defined as follows: 1) Construct Φ_{SN} as defined in Def. 12; 2) Construct φ_{SN} as defined in Def. 13; 3) Return the resulting canonical Kripke model of φ_{SN} as defined in Def. 11.*

We thus have our main theorem.

Theorem 3. *If a formula φ is satisfied in an SNM SN then φ is satisfied in the Kripke model $\mathscr{KT}(SN)$.*

5.1 Translation of Kripke Models into SNMs

Note that, in general, it is not possible to translate arbitrary Kripke models into SNMs. One of the reasons is that in Kripke models there exists only one type of predicate, which is always interpreted in the same way, whereas in SNMs, there are three types of predicates. We cannot even translate back canonical Kripke models constructed using $\mathscr{KT}$. To see why, let us consider a canonical Kripke model with the following characteristic set of formulae $\{K_{Alice}loc(Bob, library), friend(Alice, Bob)\}$. We know that the predicate $loc(Bob, library)$ belongs to Alice's knowledge base, since it is under the scope of a knowledge modality. However, we cannot know the type of the predicate $friend(Alice, Bob)$, it could be part of a connection relation, action relation or simply be a regular predicate which should appear in the environment's knowledge base.

That said, we show here that it is in fact always possible to reconstruct the original SNM from the canonical Kripke model, if we slightly modify our translation function $\mathscr{KT}$. Let Φ^m_{SN} be a *marked characteristic set*, which is a characteristic set as defined in Def. 12, but having the predicates annotated so that their type can be syntactically identified. For example, if the predicate above $friend(Alice, Bob)$ is a connection predicate, it would be converted to $co\_friend(Alice, Bob)$. We can now define $\mathscr{KT}^m$ to be a Kripke transformation function as in Def. 14, except for the input characteristic set, which is replaced by Φ^m_{SN}. Given that we can uniquely identify the type of the predicates it is trivial to define a function that takes a Kripke model constructed using $\mathscr{KT}^m$ and returns the equivalent SNM. The function proceeds as follows: firstly, it searches for all the agents present in all formulae and subformulae in Φ^m_{SN} and creates one node per agent; secondly, it puts regular predicates in the environment's knowledge base; thirdly it creates relations between agents for each connection and permission predicate; finally, for all formulae of the form $K_i\varphi$ it includes φ in i's knowledge base. We refer the reader to the extended version of this paper for the formal definitions of Φ^m_{SN}, $\mathscr{KT}^m$ and the SNM construction.

We also show that satisfaction is preserved between a marked canonical Kripke model and its original SNM when formulae are evaluated in the state corresponding to the marked characteristic set ($s_{\Phi^m_{SN}}$).

Theorem 4. *If a formula φ is satisfied in the state $s_{\Phi^m_{SN}}$ of a Kripke model $\mathscr{KT}^m(SN)$ then φ is satisfied in the SNM SN.*

6 Model checking complexity

In [5], Fagin *et al.* prove that the complexity of the model checking problem for **KD4** (without common and distributed knowledge) is PSPACE-complete for n agents where $n > 1$ and NP-complete for one agent. They also prove that for a model $M = (S, \pi, \mathscr{K}_1, \ldots, \mathscr{K}_n)$ *"There is an algorithm that, given a structure M, a state s of M and a formula* $\varphi \in \mathscr{L}$*, determines, in time* $O(||M|| \times |\varphi|)$*, whether* $(M,s) \vDash \varphi$*"* (see [5, Proposition 3.2.1]) where $||M||$ is the sum of all the states in S and the number of pairs in all $\mathscr{K}_i$, and $|\varphi|$ is the length of the formula defined as usual. This algorithm is not optimal, but the result is useful to compare the model checking problem in SNMs and the Kripke models constructed using our translation.

Let $M_{\varphi_{SN}}$ be the model $\mathscr{K}\mathscr{T}(SN)$ for an SNM SN. The complexity of the model checking problem of a formula φ in the previous model is $O(||M_{\varphi_{SN}}|| \times |\varphi|)$. $M_{\varphi_{SN}}$ has size at most $2^{|\varphi_{SN}|}$ (see Section 5), therefore it holds $||M_{\varphi_{SN}}|| \leq 2^{|\varphi_{SN}|}$. Thus, for simplicity and w.l.o.g. the above may be rewritten as $O(2^{|\varphi_{SN}|} \times |\varphi|)$.

In what follows we study the complexity of the model checking problem in $\mathscr{K}\mathscr{B}\mathscr{L}$. The proof of Theorem 1 describes an algorithm to determine whether $SN \vDash \varphi$. We consider $\mathscr{K}\mathscr{B}\mathscr{L}$ without common and distributed knowledge, since the complexity for Kripke models mentioned at the beginning of the section also excludes these modalities. For simplicity in the complexity analysis and w.l.o.g. we only consider quantifier free formulae which do not contain functions.

Let M_{KB_i} be the canonical Kripke model resulting from the conjunction of all formulae in agent's i knowledge base using our translation, the complexity of the model checking problem is given by the function *checking complexity* (**cc**): $\mathbf{cc}(p(\overrightarrow{t})) = \mathbf{cc}(c(i,j)) = \mathbf{cc}(a(i,j) = c$, $\mathbf{cc}(\neg\varphi) = 1 + \mathbf{cc}(\varphi)$, $\mathbf{cc}(\varphi_1 \wedge \varphi_2) = 1 + \mathbf{cc}(\varphi_1) + \mathbf{cc}(\varphi_2)$ and $\mathbf{cc}(K_i\varphi) = O(||M_{KB_i}|| \times |\varphi|)$ where c is an upper-bound in the cost of checking satisfaction of predicates in the environment's knowledge base, connection predicates and action predicates. Negation and conjunction need one step plus the complexity of checking satisfaction of their subformulae. Finally, satisfaction of $K_i\varphi$ depends on checking $KB_i \vdash \varphi$, which requires solving the model checking problem as defined for Kripke models. Therefore it has the same complexity. Let $outerK : \mathscr{F}_{\mathscr{K}\mathscr{B}\mathscr{L}} \rightarrow 2^{\mathscr{F}_{\mathscr{K}\mathscr{B}\mathscr{L}}}$ be a function that takes a $\mathscr{K}\mathscr{B}\mathscr{L}$ formula and returns the set of subformulae where K_i is the top most operator and it is not under the scope of a knowledge modality. For example, $outerK(K_a(p(s) \wedge K_b q(s)) \wedge p(u) \wedge \neg K_b r(s) \wedge K_c u(v)) = \{K_a(p(s) \wedge K_b q(s)), K_b r(s), K_c u(v)\}$. Note that $K_b q(s)$ is not part of the set because it is under the scope of K_a. The complexity of checking whether a formula φ is satisfiable in an SNM is $O(\sum_{K_i\varphi_i \in outerK(\varphi)}(||M_{KB_i}|| \times |\varphi_i|) + m_\varphi)$ where $m_\varphi \in \mathbb{N}$. The characteristic formula of an agent's knowledge base is the conjunction of all its knowledge, which we denote as φ_{KB_i}. As before, it holds that $||M_{KB_i}|| < |2^{\varphi_{KB_i}}|$, which we use again for the complexity of the problem $O(\sum_{K_i\varphi_i \in outerK(\varphi)}(2^{|\varphi_{KB_i}|} \times |\varphi_i|) + m_\varphi)$.

The intuition is as follows: m_φ is the cost of checking predicates, conjunctions and negations in φ, which we assume to be some constant that depends on the length of φ. Besides, $\sum_{K_i\varphi_i \in outerK(\varphi)}$ $(2^{|\varphi_{KB_i}|} \times |\varphi_i|)$ is the cost of checking each subformula φ_i in the knowledge base of the corresponding agent. In short, we have replaced checking satisfaction of φ in a complete model of the social network to checking satisfaction of subformulae of φ in the corresponding knowledge bases of the agents.

Checking the parts of φ that only contain predicates and logical connectives has very similar complexity in both models. In the canonical Kripke model of an SNM SN, the state corresponding to the characteristic set ($s_{\Phi_{SN}}$) contains all true predicates (see Def. 11). Similarly, in SNMs it is only needed to check the environment's knowledge base, and the connection and action relations (see Table 2). In both cases the complexity is determined by the length of this particular part of φ. Therefore, in order to compare the complexity of the model checking problem, we only focus on the parts of the formula that are under the scope of a knowledge modality. Given a formula φ, let φ^K be the conjunction of the subformulae starting with a K_i modality (for any $i \in Ag$), formally, $\varphi^K \triangleq \bigwedge_{\psi \in outerK(\varphi)} \psi$. Thus the complexity of the model checking problem in Kripke models is reduced to $O(2^{|\varphi_{SN}|} \times |\varphi^K|)$, and in SNMs it is $O(\sum_{K_i\varphi_i \in outerK(\varphi)} (2^{|\varphi_{KB_i}|} \times |\varphi_i|))$. To formally compare the complexity of the problem in both models we prove the following.

Lemma 8. *Given $SN \in \mathscr{SN}$ and a formula φ the following holds: $O(\sum_{K_i\varphi_i \in outerK(\varphi)} (2^{|\varphi_{KB_i}|} \times |\varphi_i|)) < O(2^{|\varphi_{SN}|} \times |\varphi^K|)$.*

The previous lemma shows that it is always more efficient to check satisfaction of a formula φ in SNMs. Intuitively, it shows that it is more efficient to construct Kripke models representing the agents' knowledge base and locally check the corresponding subformulae, than constructing the complete Kripke model to check the conjunction of the mentioned subformulae. The difference in complexity becomes more apparent as less agents are involved in the knowledge modalities of φ. When an agent is not mentioned in φ her knowledge base is disregarded. For instance, in the SNM of Fig. 2 checking $K_{Charlie}loc(Bob, pub, 1)$ requires (at most) $2^4 + 5 = 21$ steps where 4 is the size of the formula in *Charlie*'s knowledge base and 5 is the size of $K_{Charlie}loc(Bob, pub, 1)$, whereas in the corresponding canonical Kripke model it requires (at most) $2^{4+14+12} + 5 = 1073741829$ steps where 14 is the size of the conjunction of all the formulae in the knowledge base of *Alice* (assuming that the domain of x only has one element), and 12 is the size of the predicates $friend(Alice, Bob)$, $friend(Bob, Alice)$, $blocked(Bob, Charlie)$ and $friendRequest(Charlie, Alice)$.

7 Related work

The use epistemic logic to model knowledge in social networks is not new. One line of work consists in using two dimensional modal logic. It relies on Kripke models where the knowledge of the agents in the social network is encoded using an accessibility relation, and friendship is represented using a symmetric irreflexive relation between agents [14]. Other epistemic logics include a public (and private) announcement operator to study diffusion of information in the network [13, 2]. Permission and knowledge has also been merged in the so called deontic-epistemic logic [1]. For a detailed comparison among these logics and $\mathscr{KBL}$ we refer to the work by Pardo & Schneider [11, 10] and references therein.

There exist several model checkers for epistemic logic that perform efficiently in rather large scenarios [6, 3, 9]. However, as shown in this paper, model checking in the canonical Kripke model constructed from an SNM has higher complexity than in the SNM.

On the other hand, the model checking algorithm presented in this paper requires checking whether $KB_i \vdash \varphi$. As mentioned in Section 2.2, this check can be resolved by using any of the existing model checkers or SAT solvers for epistemic logic. For this reason, any improvement in the efficiency of the model checking problem in Kripke models, will also be improve the performance when checking formulae in the individual knowledge bases of each agent. In addition, local checks in different knowledge bases can easily be parallelised. For instance, if there is one process per knowledge base, formulae regarding different agents' knowledge can be checked in parallel in the corresponding knowledge bases. To the best of our knowledge, there are no parallel model checkers for epistemic logic.

8 Final Discussion

We have proved that the model checking problem in SNMs is decidable. We have shown the relation between SNMs and Kripke models. Concretely, we have proven that the belief axiomatisation **KD4**, which was originally defined for epistemic logic and naturally models agents' reasoning, is sound w.r.t. SNMs. We have provided a translation of SNMs models into canonical Kripke models and proved that satisfaction of any formula in the SNM is preserved in the corresponding Kripke model. We have also provided a translation from the canonical Kripke structure (obtained from our translation from SNMs) into the original SNM. We have proven that all formulae are satisfied in the state corresponding to the characteristic set of the SNM in the Kripke model are also satisfied in the original SNM. Finally, we showed the model checking problem in SNMs using our algorithm is more efficient than using the standard Kripke semantics.

We conjecture that arbitrary Kripke models (in the frame of models with serial and transitive relations) can be translated to SNMs. However, to preserve satisfaction the translation would generate several SNMs from a given Kripke model. Each of these SNMs would correspond to a state in the Kripke model.

The semantics of the privacy policy language $\mathscr{PPL}$ (included in $\mathscr{PPF}$) is given in terms of the satisfaction relation of $\mathscr{KBL}$, so $\mathscr{PPL}$ conformance is reduced to $\mathscr{KBL}$ satisfaction. Thanks to our results we may check conformance of $\mathscr{PPL}$ policies by using existing model checkers for epistemic logic.

Acknowledgements

This research has been supported by: the Swedish funding agency SSF under the grant *Data Driven Secure Business Intelligence* and the Swedish Research Council (*Vetenskapsrådet*) under grant Nr. 2015-04154 (*PolUser: Rich User-Controlled Privacy Policies*).

References

[1] Guillaume Aucher, Guido Boella & Leendert van der Torre (2011): *A dynamic logic for privacy compliance*. Artificial Intelligence and Law 19(2-3), pp. 187–231, doi:10.1007/s10506-011-9114-3.

[2] Zoé Christoff & Jens Ulrik Hansen (2015): *A logic for diffusion in social networks*. Journal of Applied Logic 13, pp. 48 – 77, doi:10.1016/j.jal.2014.11.011.

[3] Jan van Eijck (2007): *DEMO – A Demo of Epistemic Modelling*. Technical Report, Amsterdam University Press.

[4] Kayhan Erciyes (2014): *Complex Networks: An Algorithmic Perspective*, 1st edition. CRC Press, Inc., Boca Raton, FL, USA, doi:10.1201/b17409.

[5] Ronald Fagin, Joseph Y Halpern, Yoram Moses & Moshe Y Vardi (2003): *Reasoning about Knowledge*. The MIT press, Cambridge, MA, USA.

[6] Peter Gammie & Ron van der Meyden (2004): *MCK: Model Checking the Logic of Knowledge*. In: CAV, LNCS 3114, Springer, pp. 479–483, doi:10.1007/978-3-540-27813-9_41.

[7] Andrew K. Hirsch & Michael R. Clarkson (2013): *Belief Semantics of Authorization Logic*. In: CCS, ACM, pp. 561–572, doi:10.1145/2508859.2516667.

[8] Yabing Liu, Krishna P. Gummadi, Balachander Krishnamurthy & Alan Mislove (2011): *Analyzing Facebook Privacy Settings: User Expectations vs. Reality*. In: ACM SIGCOMM, IMC '11, ACM, pp. 61–70, doi:10.1145/2068816.2068823.

[9] Alessio Lomuscio, Hongyang Qu & Franco Raimondi (2017): *MCMAS: an open-source model checker for the verification of multi-agent systems*. STTT 19(1), pp. 9–30, doi:10.1007/s10009-015-0378-x.

[10] Raúl Pardo, Musard Balliu & Gerardo Schneider (2017): *Formalising privacy policies in social networks*. Journal of Logical and Algebraic Methods in Programming 90, pp. 125–157, doi:10.1016/j.jlamp.2017.02.008.

[11] Raúl Pardo & Gerardo Schneider (2014): *A Formal Privacy Policy Framework for Social Networks*. In: SEFM'14, LNCS 8702, Springer, pp. 378–392, doi:10.1007/978-3-319-10431-7_30.

[12] Raúl Pardo & Gerardo Schneider (2017): *Model Checking Social Network Models (Extended Version)*. Technical Report, Chalmers University of Technology. Available at `http://www.cse.chalmers.se/~pardo/papers/model-checking-SNM-full-version.pdf`.

[13] Ji Ruan & Michael Thielscher (2011): *A logic for knowledge flow in social networks*. In: *IBERAMIA*, Springer, pp. 511–520, doi:10.1007/978-3-642-25832-9_52.

[14] Jeremy Seligman, Fenrong Liu & Patrick Girard (2013): *Facebook and the Epistemic Logic of Friendship*. *CoRR* abs/1310.6440.