Model Checking Social Network Models[*]

ABSTRACT

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; iii) We show that, for SNMs, our satisfaction-based model checking algorithm is cheaper than the one based on standard Kripke models.

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

$\mathcal{PPF}$ has been specifically designed for privacy policies for real social networks, and that is why the language $\mathcal{PPL}$ and the underlying logic $\mathcal{KBL}$ 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 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 $\mathcal{PPF}$'s design (as opposed to standard Kripke models):

1. **It preserves the original structure of real social networks**. The models in $\mathcal{PPF}$ (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 $\mathcal{PPF}$ 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 $\mathcal{PPF}$ privacy policies directly depends 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

[*] The accompanying appendix includes the proofs of all theorems and lemmas and it is for reviewing purpose only.

ACM ISBN 978-1-4503-2138-9.

DOI: 10.1145/1235

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. Few 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 $\mathcal{KBL}$ is satisfied on a given SNM?[1] In this paper we study in depth the answer to these questions providing evidence that $\mathcal{PPF}$ 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 $\mathcal{KBL}$ formulae over SNMs is decidable, the algorithm being an implementation of the satisfaction relation for $\mathcal{KBL}$ (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 model checking problem for SNMs and Kripke models where we show that the former is more efficient for the Kripke models generated using our translation (Section 6).

2. PRELIMINARIES

In this section, we briefly recall First-Order Epistemic Logic [5], social network models (SNM) and the language $\mathcal{KBL}$ [11].

2.1 First-Order Epistemic Logic

We start with a set $\mathcal{T}$, consisting of *relation symbols* (p), *function symbols* (f) and *constants symbols* (c). Hereafter we will refer to $\mathcal{T}$ 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 $\mathcal{L}$, is recursively defined as follows:

DEFINITION 1 (SYNTAX OF $\mathcal{L}$, [5]).

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

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 epistemic 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*.[2]

DEFINITION 2 (RELATIONAL KRIPKE MODEL, [5]). *A* relational Kripke Model *is a tuple of the form* $\langle S, \pi, \{\mathcal{K}_i\}_{i \in Ag}\rangle$, *where:*

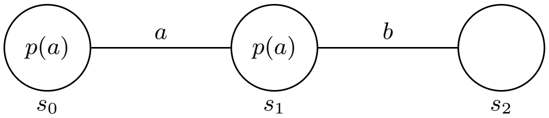

Figure 1: Relational Kripke structure

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

Table 1: Satisfaction relation over Kripke models

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

EXAMPLE 1. *Let us consider a Kripke structure consisting of the agents* a *and* b, *the states* s_0, s_1 *and* s_2, *the predicate* p *with arity 1 and the relations* $\mathcal{K}_a = \{(s_0, s_1), (s_1, s_0)\}$ *and* $\mathcal{K}_b = \{(s_1, s_2), (s_2, s_1)\}$. *For the purpose of this example we assume that all relational structures* $\pi(s_n)$ *have a common domain* $dom(\mathcal{A}) = \{a, b\}$, *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 $\mathcal{L}$ 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 3 (SEMANTICS OF $\mathcal{L}$, [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* $\varphi \in \mathcal{L}$ *to be* satisfied *by* (M, s), *written* $(M, s) \models \varphi$, *as shown in Table 1.*

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

EXAMPLE 2. *Let* M *be the model presented in Fig. 1. It holds that* $(M, s_0) \models K_a p(a)$, *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* $(M, s_1) \models \neg K_b p(a)$, *since in one of the states that* b *considers possible* $p(a)$ *is not true. In particular,* $(M, s_2) \models \neg p(a)$.

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

$\mathcal{KBL}$ is a *knowledge-based logic* for social networks. It contains all the knowledge modalities presented in $\mathcal{L}$, and additionally, it includes two special types of predicates. The connection and action

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

[2] We will sometimes omit "relational" and write "Kripke models".

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 $\mathcal{C}$ 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 4 (SYNTAX OF $\mathcal{KBL}$). *Given $i, j \in Ag$, a set of predicate symbols $\mathcal{P}$ such that $a_n(i,j), c_m(i,j), p(\overrightarrow{t}) \in \mathcal{P}$ where $m \in \mathcal{C}$ and $n \in \Sigma$, and $G \subseteq Ag$, the syntax of the* knowledge-based logic *$\mathcal{KBL}$ is inductively defined as:*

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

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 $\mathcal{L}$. $\mathcal{F}_{\mathcal{KBL}}$ denotes the set of *well-formed formulae* of $\mathcal{KBL}$ (category φ of Def. 4).

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*. Social graphs are enriched with information about the agents knowledge, permissions, connections and privacy policies as defined below.

DEFINITION 5 (SOCIAL NETWORK MODEL). *Given a set of $\mathcal{KBL}$ formulae $\mathcal{F}$, a set of privacy policies Π, and a finite set of agents $Ag \subseteq \mathcal{AU}$ from a universe $\mathcal{AU}$, a* social network model (SNM) *is a social graph of the form $\langle Ag, \mathcal{A}, KB, \pi\rangle$, where*

- *Ag is a nonempty finite set of* nodes *representing the agents of the social network.*
- *$\mathcal{A}$ is a first-order relational structure for the fixed vocabulary of the* SNM*, which as before, consists of a finite domain $dom(\mathcal{A})$*[3]*, an assignment of a k-ary relation $P^{\mathcal{A}} \subseteq dom(\mathcal{A})^{\mathcal{A}}$ for each predicate symbol, an assignment of a k-ary $f^{\mathcal{A}} : dom(\mathcal{A})^k \to dom(\mathcal{A})$ for each function symbol and and assignment of a member $c^{\mathcal{A}}$ of the domain for each constant symbol.*
- *$KB : Ag \to 2^{\mathcal{F}}$ is a function returning the set of accumulated knowledge for each agent, stored in what we call the* knowledge base *of the agent. We write KB_i to denote $KB(i)$. We assume KB_i to be consistent.*
- *$\pi : Ag \to 2^{\Pi}$ is a function returning the set of privacy policies of each agent. We write π_i for $\pi(i)$.*

The shape of the relational structure $\mathcal{A}$ depends on the type of the social network under consideration. Connections and permission actions between agents, i.e. edges of the social graph, are represented as families of binary relations, $\{C_i\}_{i\in\mathcal{C}} \subseteq Ag\times Ag$ and $\{A_i\}_{i\in\Sigma} \subseteq Ag \times Ag$ 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. $\mathcal{SN}$ denotes the universe of all possible SNMs.

The knowledge base KB_i contains the explicit knowledge that agent i has. Besides this explicit knowledge, agents also know anything that can be derived from formulae in their knowledge bases (using the **S5** axiomatisation of epistemic logic [5]). The function $Cl : 2^{\mathcal{F}_{\mathcal{KBL}}} \to 2^{\mathcal{F}_{\mathcal{KBL}}}$ computes such inferences.

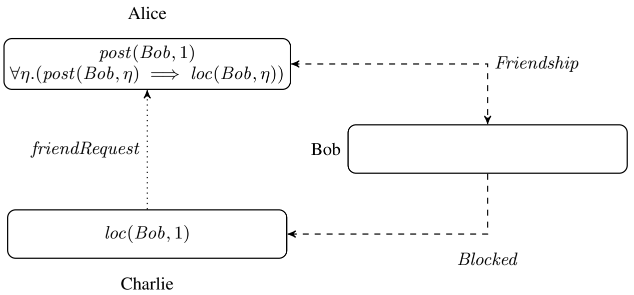

Figure 2: Example of Social Network Model

DEFINITION 6 (CLOSURE OF A KNOWLEDGE BASE). *The* knowledge closure *of agent i, $Cl(KB_i)$, satisfies the following:*
a) If $\varphi \in KB_i$, then $\varphi \in Cl(KB_i)$;
b) For all $\varphi \in \mathcal{F}_{\mathcal{KBL}}$, if $\varphi \in Cl(KB_i)$ then $\neg\varphi \notin Cl(KB_i)$;
c) $\varphi \wedge \psi \in Cl(KB_i)$ iff $\varphi \in Cl(KB_i)$ and $\psi \in Cl(KB_i)$;
d) If $\varphi \in Cl(KB_i)$ and $\varphi \implies \psi \in Cl(KB_i)$, then $\psi \in Cl(KB_i)$;
e) If φ is probable in **S5***, then $\varphi \in Cl(KB_i)$;*
f) If $\varphi \in Cl(KB_i)$ then $K_i\varphi \in Cl(KB_i)$;
g) If $\varphi \notin Cl(KB_i)$ then $\neg K_i\varphi \in Cl(KB_i)$.

Note that clauses a) - e) of Cl effectively correspond to the standard definition of an **S5** maximal consistent set as defined by Fagin *et al.* [5, Lemma 3.1.2]. Additionally, we include clauses f) and g), which make agents aware of their knowledge (both what they know and what they do not know). We use these clauses to encode axioms A4 and A5 in social network models (see Section 4). More importantly, the problem of checking whether $\varphi \in Cl(KB_i)$ is equivalent to $KB_i \vdash \varphi$[4] (since $Cl(KB_i)$ is an **S5** maximal consistent set [5, Theorem 3.2.4]) which can be solved by using any existing theorem prover for epistemic logic (e.g. [6, 9, 10, 16]) and is a decidable problem [5, Collorary 3.2.3].

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 Friendship and Blocked, $\mathcal{C} = \{Friendship, Blocked\}$. Here, we define $dom(\mathcal{A})$ to be a finite set of post ids.*

Fig. 2 shows a graphical representation of SN. In this model the dashed arrows represent connections. Note that the Friendship 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. Charlie and Bob have $loc(Bob, 1)$ in their knowledge bases, meaning that both know location nr. 1 of Bob. Agents' nodes can also contain more complex $\mathcal{KBL}$ formulae that may increase their knowledge. For instance, Alice knows $loc(Bob, 1)$ implicitly. Alice can in fact derive that by Modus Ponens*, from $post(Bob, 1)$ and $\forall\eta.post(Bob,\eta) \implies loc(Bob,\eta)$.*

In what follows we define the satisfaction relation for $\mathcal{KBL}$ formulae, interpreted over social network models.

[3] For the sake of simplicity, 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 as all our results will still hold. For instance, we can have $dom(\mathcal{A})$ consisting of the domain of agents, indexes for posts, indexes for pictures, etc.

[4] We use $KB_i \vdash \varphi$ to denote that from the conjunction of all formulae in KB_i, φ can be derived using the axioms and derivation rules of **S5** (see Appendix A), formally, $\bigwedge_{\psi\in KB_i}\psi \vdash \varphi$.

$SN \models p(\overrightarrow{t})$	iff	$p(\overrightarrow{t}) \in KB_e$
$SN \models c_m(i,j)$	iff	$(i,j) \in C_m$
$SN \models a_n(i,j)$	iff	$(i,j) \in A_n$
$SN \models \neg\varphi$	iff	$SN \not\models \varphi$
$SN \models \varphi \wedge \psi$	iff	$SN \models \varphi$ and $SN \models \psi$
$SN \models \forall x.\varphi$	iff	for all $v \in dom(\mathcal{A})$, $SN \models \varphi[v/x]$
$SN \models K_i\varphi$	iff	$\varphi \in Cl(KB_i)$

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

DEFINITION 7 (SEMANTICS OF $\mathcal{KBL}$). *Given an SNM* $SN = \langle Ag, \mathcal{A}, KB, \pi\rangle$*, agents* $i, j \in Ag$*, formulae* $\varphi, \psi \in \mathcal{F}_{\mathcal{KBL}}$*, a finite set of agents* $G \subseteq Ag$*,* $m \in \mathcal{C}$ *and* $n \in \Sigma$ *the* satisfaction relation $\models \subseteq \mathcal{SN} \times \mathcal{KBL}$ *is defined in Table 2.*

The intuition behind the semantic definition of the knowledge modality is different in $\mathcal{KBL}$ 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 $p(\overrightarrow{t})$. The environment's knowledge base (KB_e) contains all predicates which are true in the real world. For instance, $location(Alice) = \text{“}Sweden''$ or $Alice \notin \{Bob, Charlie\}$. For simplicity, sometimes we use predicates to just represent pieces of information and these predicates are always present in KB_e. For instance, in Example 3, we use the predicate $loc(Bob, 1)$ to represent location 1 of Bob, which allows us to write formulae such as $K_{Alice}loc(Bob, 1)$ to state that "Alice knows location 1 of Bob".

EXAMPLE 4. *Let SN be the SNM in Fig. 2. As described in Example 3, Alice knows post 1 of Bob, meaning that* $SN \models K_{Alice}post(Bob, 1)$ *holds. Indeed, it holds since* $post(Bob, 1)$ *is in the knowledge base of Alice, i.e.,* $post(Bob, 1) \in KB_{Alice}$ *(1).*

Though not explicitly stated, it is possible for Alice to derive location 1 of Bob, meaning that $SN \models K_{Alice}loc(Bob, 1)$ *(2) should hold. Following the semantics of* K_i *in Table 2, the previous formula is true iff* $loc(Bob, 1) \in Cl(KB_{Alice})$*. Fig. 2 shows that* KB_{Alice} *contains the formula* $\forall \eta.post(Bob, \eta) \implies loc(Bob, \eta)$ *(3) where* $\eta \in \mathbb{N}$*, therefore the deductive engine derives* $post(Bob, 1) \implies loc(Bob, 1)$ *(4).*

From (1) and (4), by modus ponens *we can derive* $loc(Bob, 1)$*, i.e.* $loc(Bob, 1) \in Cl(KB_{Alice})$*, hence (2) holds.*

3. MODEL CHECKING SNMS

Here we present a model checking algorithm that directly implements the semantics of $\mathcal{KBL}$ in Table 2, and we show that it is decidable under the following reasonable assumptions:

ASSUMPTION 1. *All domains are finite.*

ASSUMPTION 2. *All functions are computable.*

These assumptions are present in all real social network. 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 $\mathcal{KBL}$ terms must be computable. As mentioned in the introduction, $\mathcal{KBL}$ 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 concrent social network. For instance, in [12] Pardo & Schneider 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* $\varphi \in \mathcal{F}_{\mathcal{KBL}}$ *be a formula. Determining whether* $SN \models \varphi$ *is decidable.*

PROOF. We enumerate the sequence of steps of a model checking algorithm and show that all of them are decidable:

1. We expand the universal quantifiers in φ by inductively transforming each subformula $\forall x.\varphi'$ into a conjunction with one cojunct $\varphi'[v/x]$ for each element v of the domain $dom(\mathcal{A})$. Given that the domain is finite (see Assumption 1), it always terminates and results in a a quantifier free formula.
2. We compute all functions and replace all constants with an element of the domain according to the assignment in $\mathcal{A}$. 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.
3. We now inductively show that all the elements of the formula (see Definition 4) can be computed.
   - Checking $c_m(i,j)$ and $a_n(i,j)$ can be performed in constant time, simply by checking the model.
   - Checking $p(\overrightarrow{t})$ requires a query to the environment's knowledge base KB_e, which 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 $\varphi \in Cl(KB_i)$. This query is equivalent to checking derivability from the set of formulae in the agent's knowledge base, i.e, $KB_i \vdash \varphi$ (see Section 6), which is a decidable problem [5].

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

It is easy to see that the semantics of $\mathcal{KBL}$ is captured by the algorithm. □

In Section 6 we study the computational 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

In this section we explore some 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 Cl. As we described in Def. 6, Cl includes all the axioms and derivation rules from **S5**.

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 $friend(Bob, Charlie) \in Cl(KB_{Alice})$. 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* and i is an agent: D. $\neg K_i \perp$.

In Kripke models, axiom D is present when the accessibility relation is serial [5]. In SNMs, the knowledge bases of the agents and Cl are consistent by definition. Therefore, $\perp$ cannot be derived.

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

As we mentioned in the introduction, $\mathcal{KBL}$ and SNMs were developed in the context of a privacy policy framework for social networks [12, 11]. In a 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 $\varphi \in \mathcal{L}$ 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 $\varphi \in \mathcal{L}$ 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. *Axioms A4 and A5 are sound with respect to SNMs.*

The axiomatisation **K** together with axioms D, A4 and A5 forms the so-called **KD45** axiomatisation. We thus have the following result for SNMs.

THEOREM 2. ***KD45** 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 $\varphi \in \mathcal{L}$ and a set of agents G, common knowledge is defined as follows:

$$(M, s) \models C_G\varphi \text{ iff } (M, s) \models E^k_G\varphi \text{ for } k = 1 \ldots$$

where $E^0_G\varphi = \varphi$ and $E^{k+1}_G\varphi = E_G\varphi E^k_G\varphi$. The definition of common knowledge for SNMs is analogous to the one above.

DEFINITION 8 (COMMON KNOWLEDGE). *Given an SNM SN, a formula $\varphi \in \mathcal{F}_{\mathcal{KBL}}$ and a set of agents G, common knowledge is defined as follows:*

$$SN \models C_G\varphi \textit{ iff } SN \models E^k_G\varphi \textit{ for } k = 1 \ldots$$

Given formulae $\varphi, \psi \in \mathcal{L}$, 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 4. *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,

$$(M, s) \models D_G\varphi \text{ iff } (M, t) \models \varphi \text{ for all } t \text{ such that } (s, t) \in \bigcap_{i \in G} \mathcal{K}_i.$$

We define distributed knowledge as the union of all the explicit knowledge that all the agents in G has and everything that can be derived from it.

DEFINITION 9 (DISTRIBUTED KNOWLEDGE). *Given an SNM SN, a formula $\varphi \in \mathcal{F}_{\mathcal{KBL}}$ and a set of agents G, distributed knowledge is defined as follows:*

$$SN \models D_G\varphi \textit{ iff } \varphi \in Cl(\bigcup_{i \in G} KB_i).$$

The following axioms characterise distributed knowledge [5]:

D1. $D_{\{i\}}\varphi \iff K_i\varphi,\ i = 1, \ldots, n$,
D2. $D_G\varphi \implies D_{G'}(\varphi)$ if $G \subseteq G'$,
DA2-DA5. Axioms A2, A4 and A5 of **KD45**, replacing the modality K_i with the modality D_G for each axiom.

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

LEMMA 5. *Axioms D1 and D2, together with the axioms A2, A4 and A5 of the* ***KD45****-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 $\mathcal{KBL}$ 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 ***KD45****-consistent* if $\neg\varphi$ cannot be derived. A set of formulae is **KD45**-consistent if the conjunction of all the formulae in the set is **KD45**-consistent. We say that a set of formulae Φ is *maximal* ***KD45****-consistent* with respect to the language $\mathcal{L}$, if Φ is **KD45**-consistent and for all φ in $\mathcal{L}$ but not in Φ, the set $\Phi \cup \{\varphi\}$ is not **KD45**-consistent. In what follows, we describe the procedure of how to construct a canonical Kripke model for a **KD45**-consistent formula. We will follow a similar approach when translating SNMs into Kripke models.

DEFINITION 10 (CANONICAL KRIPKE MODEL FOR **KD45**). *Consider a* **KD45**-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* ***KD45****-consistent subsets of $Sub^+(\varphi)$. Given a set of formulae $\Theta \subseteq \mathcal{L}$, we define $\Theta/K_i = \{\varphi \mid K_i\varphi \in \Theta\}$. The canonical Kripke model for φ is defined as follows: $M_\varphi = \langle S_\varphi, \pi, \{\mathcal{K}_i\}_{i\in Ag}\rangle$ where*

- $S_\varphi = \{s_\Theta \mid \Theta \in Con(\varphi)\}$
- $\mathcal{K}_i = \{(s_\Theta, s_\Psi) \mid \Theta/K_i = \Psi/K_i,\ \Theta/K_i \subseteq \Psi\}$
- $\pi(s_\Theta)(p(t_1,\ldots,t_k)) = \begin{cases} \textbf{\textit{true}} & \text{if } p(t_1,\ldots,t_k) \in \Theta \\ \textbf{\textit{false}} & \text{if } p(t_1,\ldots,t_k) \notin \Theta \end{cases}$

The formula φ will be satisfied in the resulting canonical Kripke model [5, Theorem 3.2.4]. The set of Kripke models that are sound and complete with respect to **KD45** are the ones with an Euclidean, serial and transitive accessibility relation. The accessibility relation of the previous canonical Kripke model is, as shown in [5, Theorem 3.2.4], Euclidean, serial and transitive. We denote the set of Kripke models with this accessibility relation as $\mathcal{M}^{elt}$.

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 converting an SNM to 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 in an SNM increases monotonically, i.e., the users 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 11 (CHARACTERISTIC SET). *The* characteristic set *of an SNM SN, denoted as Φ_{SN}, is constructed as follows, $\Phi_{SN} = \{p(\overrightarrow{t}) \mid p(\overrightarrow{t}) \in KB_e\} \cup \{K_i\varphi \mid \varphi \in KB_i\} \cup \{c(i,j) \mid (i,j) \in C_c, c \in \mathcal{C}\} \cup \{a(i,j) \mid (i,j) \in A_a, a \in \Sigma\}$.*

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

DEFINITION 12 (CHARACTERISTIC FORMULA). *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 **KD45**-consistent.

LEMMA 6. *For all $SN \in \mathcal{SN}$, φ_{SN} is* ***KD45****-consistent.*

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

DEFINITION 13 (KRIPKE TRANSFORMATION FUNCTION). *Let $\mathcal{KT} : \mathcal{SN} \to \mathcal{M}^{elt}$ be a function which takes an SNM and converts it to the corresponding Kripke model as follows. Given an $SN \in \mathcal{SN}$, $\mathcal{KT}(SN)$ is defined as follows: 1) Construct Φ_{SN} as defined in Def. 11; 2) Construct φ_{SN} as defined in Def. 12; 3) Return the resulting canonical Kripke model of φ_{SN} as defined in Def. 10.*

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 $\mathcal{KT}(SN)$.*

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 $\mathcal{KT}$. To see why let us consider a canonical Kripke model with the following characteristic set of formulae $\{K_{Alice}loc(Bob), friend(Alice, Bob)\}$. We know that the predicate $loc(Bob)$ 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 $\mathcal{KT}$. Let Φ^m_{SN} be a *marked characteristic set*, which is a characteristic set as defined in Def. 11, 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 $\mathcal{KT}^m$ to be a Kripke transformation function as in Def. 13, 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 $\mathcal{KT}^m$ 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. See

Appendix D for the formal definitions of Φ^m_{SN}, $\mathcal{KT}^m$ and the SNM construction).

Furthermore, it is easy to 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 $\mathcal{KT}^m(SN)$ 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 **KD45** (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 $M = (S, \pi, \mathcal{K}_1, \dots, \mathcal{K}_n)$ *"There is an algorithm that, given a structure M, a state s of M and a formula $\varphi \in \mathcal{L}$, determines, in time $O(||M|| \times |\varphi|)$, whether $(M, s) \models \varphi$"* (see [5, Proposition 3.2.1]) where $||M||$ is the sum of all the states in S and the number of pairs in all $\mathcal{K}_i$, 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 $\mathcal{KT}(SN)$ 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 at most $2^{|\varphi_{SN}|}$ (see Section 5), therefore it holds $||M_{\varphi_{SN}}|| \geq 2^{|\varphi_{SN}|}$. 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 $\mathcal{KBL}$. The proof of Theorem 1 describes an algorithm to determine whether $SN \models \varphi$. We consider $\mathcal{KBL}$ 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**):

$$\begin{array}{rcl@{\qquad}rcl}
\mathbf{cc}(p(\overrightarrow{t})) & = & c & \mathbf{cc}(\neg\varphi) & = & 1 + \mathbf{cc}(\varphi) \\
\mathbf{cc}(c(i,j)) & = & c & \mathbf{cc}(\varphi_1 \wedge \varphi_2) & = & 1 + \mathbf{cc}(\varphi_1) + \mathbf{cc}(\varphi_2) \\
\mathbf{cc}(a(i,j)) & = & c & \mathbf{cc}(K_i\varphi) & = & O(||M_{KB_i}|| \times |\varphi|)
\end{array}$$

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 $\varphi \in Cl(KB_i)$, which requires solving the model checking problem as defined for Kripke models. Therefore it has the same complexity. Let $outerK : \mathcal{F}_{\mathcal{KBL}} \to 2^{\mathcal{F}_{\mathcal{KBL}}}$ be a function that takes a $\mathcal{KBL}$ 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. 10). 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 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 7. *Given $SN \in \mathcal{SN}$ 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 $K_{Charlie}\, loc(Bob, 1)$ requires (at most) $2^3 + 4 = 12$ steps where 3 is the size of the the formula in $Charlie$'s knowledge base and 4 is the size of $K_{Charlie}\, loc(Bob, 1)$, whereas in the corresponding canonical Kripke model it requires (at most) $2^{3+11+12} + 4 = 268435456$ steps where 11 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 [15, 14]. Other epistemic logics include a public (and private) announcement operator to study diffusion of information in the network [13, 3, 2]. Permission and knowledge has also been merged in the so called deontic-epistemic logic [1]. For a detailed comparison among these logics and $\mathcal{KBL}$ we refer to the work by Pardo & Schneider [12, 11] and references therein.

There exist several model checkers for epistemic logic that perform efficiently in rather large scenarios [6, 16, 10]. 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 $\varphi \in Cl(KB_i)$. 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 presented a decidable algorithm for the model checking problem in SNMs. We have shown the relation between SNMs and Kripke models. Concretely, we have proven that the **KD45** axiomatisation originally defined for epistemic logic 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 Euclidean, serial and transitive relations) can be translated to SNMs. However, in order 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 $\mathcal{PPL}$ (included in $\mathcal{PPF}$) was given in terms of the satisfaction relation of $\mathcal{KBL}$. Thanks to the results in this paper, we could check satisfaction of $\mathcal{PPL}$ policies over Kripke models generated from an SNM (by using some of the existing model checkers for epistemic logic). That said and given our complexity result, it will be more efficient to implement an *ad hoc* model checker for social networks implementing the algorithm presented in this paper, together with existing epistemic logic solvers for deciding local knowledge for each agent. We leave as future work a quantitative study to precisely determine the performance improvement of using our approach as opposed to standard Kripke models.

9. REFERENCES

[1] Guillaume Aucher, Guido Boella, and Leendert Torre. A dynamic logic for privacy compliance. *Artificial Intelligence and Law*, 19(2-3):187–231, 2011.

[2] Zoé Christoff and Jens Ulrik Hansen. Dynamic social networks logic. *ILLC Prepublication Series Report PP-2014-09*, 2014.

[3] Zoé Christoff and Jens Ulrik Hansen. A logic for diffusion in social networks. *Journal of Applied Logic*, 13:48 – 77, 2015.

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

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

[6] Peter Gammie and Ron van der Meyden. Mck: Model checking the logic of knowledge. In *CAV*, volume 3114 of *LNCS*. Springer, 2004.

[7] Andrew K. Hirsch and Michael R. Clarkson. Belief semantics of authorization logic. In *CCS*, pages 561–572. ACM, 2013.

[8] Yabing Liu, Krishna P. Gummadi, Balachander Krishnamurthy, and Alan Mislove. Analyzing facebook privacy settings: User expectations vs. reality. In *ACM SIGCOMM*, IMC '11, pages 61–70. ACM, 2011.

[9] Alessio Lomuscio, Hongyang Qu, and Franco Raimondi. Mcmas: A model checker for the verification of multi-agent systems. In *CAV*, pages 682–688. Springer, 2009.

[10] Alessio Lomuscio, Hongyang Qu, and Franco Raimondi. Mcmas: an open-source model checker for the verification of multi-agent systems. *International Journal on Software Tools for Technology Transfer*, pages 1–22, 2015.

[11] Raúl Pardo. *Formalising Privacy Policies for Social Networks*. Department of Computer Science and Engineering, Chalmers University of Technology, 2015. Pages 102. Licentiate thesis.

[12] Raúl Pardo and Gerardo Schneider. A formal privacy policy framework for social networks. In *SEFM'14*, volume 8702 of *LNCS*, pages 378–392. Springer, 2014.

[13] Ji Ruan and Michael Thielscher. A logic for knowledge flow in social networks. In *IBERAMIA*, pages 511–520. Springer, 2011.

[14] Jeremy Seligman, Fenrong Liu, and Patrick Girard. Facebook and the epistemic logic of friendship. In *TARK*, 2013.

[15] Jeremy Seligman, Fenrong Liu, and Patrick Girard. Knowledge, friendship and social announcement. In *Logic Across the University: Foundations and Applications*, 2013.

[16] Jan van Eijck. Demo – a demo of epistemic modelling. In *7th ADMW*, volume 1, page 303, 2007.

APPENDIX

A. AXIOMATISATION S5

Axioms

(A1) All (instances of) first-order tautologies

(A2) $(K_i\varphi \wedge K_i(\varphi \implies \psi)) \implies K_i\psi$

(A3) $K_i\varphi \implies \varphi$

(A4) $K_i\varphi \implies K_iK_i\varphi$

(A5) $\neg K_i\varphi \implies K_i\neg K_i\varphi$

Derivation rules

(Modus Ponens) $\dfrac{\varphi \quad \varphi \implies \psi}{\psi}$

(Necessitation) $\dfrac{\varphi}{K_i\varphi}$ where φ must be provable from no assumptions.

B. SOUNDNESS PROOFS

Soundness of axiom D

LEMMA 1. *For all $SN \in \mathcal{SN}$ and some agent i the following holds:*

$$SN \models \neg K_i\bot.$$

PROOF. Given that KB_i is consistent (see Def. 5) and by clause b) in the definition of Cl (Def. 6), it trivially follows that $\bot$ cannot be derived. □

Soundness of K-axiomatisation

LEMMA 2. *For all formulae φ and ψ in $\mathcal{F}_{\mathcal{KBL}}$, $SN \in \mathcal{SN}$ and some agent i the following holds:*

a) A1. All (instances of) first-order tautologies,

b) A2. $SN \models (K_i\varphi \wedge K_i(\varphi \implies \psi)) \implies K_i\psi$,

c) R1. From $SN \models \varphi$ and $SN \models \varphi \implies \psi$ infer $SN \models \psi$,

d) R2. From $\models \varphi$ infer $\models K_i\varphi$ where φ must be provable from no assumptions.

PROOF.

a) It follows from the definition of the environment's knowledge base. KB_e is defined to include all truth, which includes all tautologies of FOEL.

b) It trivially follows from clause c) of the definition of Cl (Def. 6).

c) It follows immediately from the fact that the interpretation of $\wedge$ and $\neg$ in the definition of $\models$ is the same as in First-Order Logic.

d) It follows from clause e) of the definition of Cl (Def. 6). $\models K_i\varphi$ holds iff $\varphi \in Cl(KB_i)$ for all SNMs. Let φ be any formula provable from no assumptions, then it holds that $\models \varphi$. By clause e) we can use necessitation to deduce that $K_i\varphi \in Cl(KB_i)$. Finally, from clause e) by using A3 we derive that $\varphi \in Cl(KB_i)$ as desired. □

Soundness of axioms A4 and A5

LEMMA 3. *For all formula φ in $\mathcal{F}_{\mathcal{KBL}}$, $SN \in \mathcal{SN}$ and some agent i the following holds:*

a) A4. $SN \models K_i\varphi \implies K_iK_i\varphi$,

b) A5. $SN \models \neg K_i\varphi \implies K_i\neg K_i\varphi$.

PROOF.

a) It follows from clause d) of the definition of Cl (Def. 6). Consider an agent i that knows a formula φ, i.e., $K_i\varphi$. Then $\varphi \in Cl(KB_i)$. Finally, from the definition of Cl (Def. 6) we can conclude that $K_i\varphi \in Cl(KB_i)$ also holds, meaning that $SN \models K_iK_i\varphi$.

b) It follows from clause e) of the definition of Cl (Def. 6). Consider an agent i that does not know a formula φ, i.e., $\neg K_i\varphi$. Then $\varphi \notin Cl(KB_i)$. Since $K_i\varphi$ is not in $Cl(KB_i)$ either, by clause e) we can derive that $\neg K_i\varphi \in Cl(KB_i)$. Finally, by definition of $\models$ conclude that $SN \models K_i\neg K_i\varphi$ holds. □

Soundness of KD45-axiomatisation

THEOREM 2. *The **KD45** axiomatisation is sound with respect to SNMs.*

PROOF. It follows from lemmas 1, 2 and 3. □

Soundness of the axiomatisation of common knowledge

LEMMA 4. *For all formula φ written in $\mathcal{F}_{\mathcal{KBL}}$, $SN \in \mathcal{SN}$ and some group of agents G the following holds:*

a) C1. $SN \models E_G\varphi \iff \bigwedge_{i\in G} K_i\varphi$,

b) C2. $SN \models C_G\varphi \iff E_G(\varphi \wedge C_G\varphi)$,

c) RC1. From $\models \varphi \implies E_G(\varphi\wedge\psi)$ infer $\models \varphi \implies C_G\psi$ where $\varphi \implies E_G(\psi \wedge \varphi)$ must be provable from no assumptions.

PROOF.

a) It trivally follows from the definition of E_G.

b) It follows from the definition of common knowledge. First let us prove the implication from right to left. Assume that $SN \models C_G\varphi$. It means that $E_G\varphi \wedge E_GE_G\varphi \wedge E_GE_GE_G\varphi \wedge \ldots \in Cl(KB_i)$ for all $i \in G$. By distributivity of E_G we can derive that $E_G(\varphi \wedge E_GE_G\varphi \wedge E_GE_G\varphi \wedge \ldots) \in Cl(KB_i)$. Hence $\varphi \in Cl(KB_i)$ and $E_G\varphi \wedge E_GE_G\varphi \wedge \ldots \in Cl(KB_i)$. Finally, by the definition of C_G, we conclude that $\varphi \wedge C_G\varphi \in Cl(KB_i)$ for all $i \in G$, thus $SN \models E_G(\varphi \wedge C_G\varphi)$. The other direction of the implication follows directly from the definition of K_i and C_G. Assume $SN \models E_G(\varphi \wedge C_G\varphi)$. It means that for all $i \in G$, $C_G\varphi \in Cl(KB_i)$, hence by the definition of C_G we know that $\varphi \wedge E_G\varphi \wedge E_GE_G\varphi \wedge \ldots \in Cl(KB_i)$, which by definition of $\models$ means that $SN \models E_G(\varphi \wedge E_G\varphi \wedge E_GE_G\varphi \wedge \ldots)$. Finally, by distributivity of E_G and definition of C_G we conclude that $SN \models C_G\varphi$.

c) It follows from clause d) of the definition of Cl (Def. 6). $\varphi \implies E_G(\varphi \wedge \psi)$ is true in all SNMs, which means that it is derivable from no assumptions. The axiomatisation **S5** also includes RC1, therefore from $\varphi \implies E_G(\varphi \wedge \psi)$ it can be derived that $\models \varphi \implies C_G\psi$ for all agents. Finally, since we made no assumption of any concrete SNM, we conclude that $\models \varphi \implies C_G\psi$. □

Soundness of the axiomatisation of distributed knowledge

LEMMA 5. *For all formula φ in $\mathcal{F}_{\mathcal{KBL}}$, $SN \in \mathcal{SN}$ and some group of agents G, the following holds:*

a) D1. $SN \models D_{\{i\}}\varphi \iff K_i\varphi,\ i = 1, \ldots, n$,

b) D2. $SN \models D_G\varphi \implies D_{G'}(\varphi)$ if $G \subseteq G'$,

*c) DA2-DA5. Axioms A2, A4 and A5 of **KD45**, replacing the modality K_i with the modality D_G for each axiom.*

PROOF.

a) It easily follows from the definition of distributed knowledge. $SN \models D_{\{i\}}\varphi$ iff $\varphi \in Cl(\bigcup_{j\in\{i\}} KB_j)$ and $Cl(\bigcup_{j\in\{i\}} KB_j) = Cl(KB_i)$. Hence we can conclude that $D_{\{i\}}\varphi$ iff $\varphi \in Cl(KB_i)$. By definition of $\models$, we know that $K_i\varphi$ iff $\varphi \in Cl(KB_i)$. Therefore, we can conclude that $SN \models D_{\{i\}}\varphi \iff K_i\varphi$.

b) By the definition of distributed knowledge we know that $SN \models D_{G'}\varphi$ holds if and only if $\varphi \in Cl(\cup_{i\in G'} KB_i)$, since $G \subseteq G'$, we can rewrite the previous expression as $\varphi \in Cl((\cup_{i\in G} KB_i) \cup (\cup_{i\in G'\setminus G} KB_i))$. We know from the assumption that $D_G\varphi$ holds, hence $\varphi \in Cl(\cup_{i\in G} KB_i)$. By Lemma 8, we can deduce that $\varphi \in Cl((\cup_{i\in G} KB_i) \cup (\cup_{i\in G'\setminus G} KB_i))$ holds. Finally, by the definition of $\models$, we can conclude that $SN \models D_{G'}\varphi$.

c) Distributed knowledge is equivalent to an agent which contains all the knokwledge of the members of the group. Therefore, the proofs of all the axioms and derivation rules is analogous to those of Lemmas 2 and 3, just by replacing K_i with D_G. □

LEMMA 8. *Given the set formulae $\Phi \subseteq \mathcal{F}_{\mathcal{KBL}}$, two formulae $\varphi, \psi \in \mathcal{F}_{\mathcal{KBL}}$ and the closure function Cl the following holds:*

$$\text{If } \varphi \in Cl(\Phi) \text{ then } \varphi \in Cl(\Phi \cup \{\psi\}).$$

PROOF. If Φ or $\Phi \cup \{\psi\}$ are inconsistent, then anything can be derived after applying Cl (including φ), therefore if $\Phi \cup \{\psi\}$ is inconsistent, $\varphi \in Cl(\Phi \cup \{\psi\})$.

If $\Phi \cup \{\psi\}$ is consistent, all formulae in $Cl(\Phi)$ will still remain in $Cl(\Phi \cup \{\psi\})$, since none of the axioms that Cl applies (the axiomatisation **S5**) to the formulae in Φ remove any formula (see [5]). Therefore, if $\varphi \in Cl(\Phi)$ then we can conclude that $\varphi \in Cl(\Phi \cup \{\psi\})$. □

C. PRESERVATION OF SATISFIABILITY OF A CANONICAL KRIPKE MODEL CONSTRUCTED FROM AN SNM

THEOREM 3. *If a formula φ is satisfiable in an SNM SN then φ is satisfiable in the Kripke model $\mathcal{KT}(SN)$.*

PROOF. Fagin *et al.* have shown that the canonical Kripke model which satisfies a **KD45**-consistent formula (and everything that can be derived from it) can be constructed [5]. Therefore, if a formula φ is in the maximal **KD45**-consistent set used to construct the Kripke model it will be satisfiable in the model. Their proof can be adapted to our case. In particular, we show that if a formula φ is satisfiable in SN, then it is included in the maximal **KD45**-consistent set $\mathcal{MC}(\Phi_{SN})$, where $\mathcal{MC}$ is a function which extends a **KD45**-consistent set to the corresponding maximal **KD45**-consistent one. So, we will prove that for any formula φ, if $SN \models \varphi$ then $\varphi \in \mathcal{MC}(\Phi_{SN})$ and by [5, Theorem 3.2.4] it holds that φ is satisfiable in the corresponding canonical Kripke model defined in Def. 13. The proof is carried out by induction on the structure of the formula.

Case $\varphi = p(\vec{t})$. Assume that $SN \models p(\vec{t})$ holds, then $p(\vec{t}) \in KB_e$ and by Def 13 $p(\vec{t}) \in \Phi_{SN}$.

Case $\varphi = a(i,j)$ or $\varphi = c(i,j)$. Assume that $SN \models a(i,j)$ holds, then $a(i,j) \in A_a$ and by Def 13 $a(\vec{t}) \in \Phi_{SN}$. The exact same proof holds for $c(i,j)$.

Case $\varphi = \neg\psi$. Assume by contradiction that $SN \models \neg\psi$ holds and $\psi \in \mathcal{MC}(\Phi_{SN})$. As shown in Theorem 3, if $\psi \in \mathcal{MC}(\Phi_{SN})$ then $SN \models \psi$, which leads to a contradiction, and thus $\psi \notin \mathcal{MC}(\Phi_{SN})$ holds. By definition of $\mathcal{MC}$ if $\psi \notin \mathcal{MC}(\Phi_{SN})$ then $\neg\psi \in \mathcal{MC}(\Phi_{SN})$ as required. Note that this argument is not circular, since Theorem's 3 proof does not depend on the proof of this Theorem.

Case $\varphi = \psi_1 \wedge \psi_2$. By induction hypothesis assume that If $SN \models \psi_1$ then $\psi_1 \in \mathcal{MC}(\Phi_{SN})$ (and similarly for ψ_2). By definiton of $\models$, it follows that $SN \models \psi_1 \wedge \psi_2$ holds iff $SN \models \psi_1$ and $SN \models \psi_1$. Finally by induction hypothesis we conclude that $\psi_1 \in \mathcal{MC}(\Phi_{SN})$ and $\psi_2 \in \mathcal{MC}(\Phi_{SN})$ holds.

Case $\varphi = \forall x.\psi$. By induction hypothesis assume that if $SN \models \psi[v/x]$ then $\psi[v/x] \in \mathcal{MC}(\Phi_{SN})$. By definition of $\models$ it follows that $SN \models \forall x.\psi$ iff $SN \models \psi[v/x]$ for all $v \in dom(\mathcal{A})$. Finally, we conclude by induction hypothesis that $\psi[v/x] \in \mathcal{MC}(\Phi_{SN})$.

Case $\varphi = K_i\psi$. Assume that $SN \models K_i\psi$. By definition of $\models$ it holds that $\psi \in Cl(KB_i)$. It leads to two possible cases: 1) If $\psi \in KB_i$ then $\psi \in \Phi_{SN}$ by Def. 13; 2) if $\psi \in Cl(KB_i) \setminus KB_i$ then ψ has been derived therefore $\psi \in \mathcal{MC}(\Phi_{SN})$ again by Def. 13.

Case $\varphi = S_G\psi$ or $\varphi = E_G\psi$. Both are derived operators from K_i. The proof easily follows by their definition and the proof provided for the case $\varphi = K_i\psi$.

Case $\varphi = C_G\psi$. Assume that $SN \models C_G\psi$ holds. By axiom C2 and the definition $\models$ we can derive that $\forall j \in G\ (\psi \wedge C_G\psi) \in KB_j$. By Def. 13 and $\models$ it also holds that $E_G(\psi \wedge C_G\psi) \in \mathcal{MC}(\Phi_{SN})$. Finally, we can derive using axiom C2 that $C_G\psi \in \mathcal{MC}(\Phi_{SN})$ as required.

Case $\varphi = D_G\psi$. Let $DKB_G \triangleq \cup_{i\in G} KB_i$. Assume that $SN \models D_G\psi$. By definition of $\models$ it holds that $\psi \in Cl(DKB_G)$. It leads to two possible cases: 1) If $\psi \in DKB_G$ then there exist an agent $i \in G$ such that $\psi \in KB_i$ therefore $\psi \in \Phi_{SN}$ by Def. 13. 2) If $\psi \in Cl(DKB_G) \setminus DKB_G$ then it is derived from the explicit knowledge of the agents using any of the axioms in **KD45** (including common knowledge and distributed knowledge). Therefore we can conclude that $\psi \in \mathcal{MC}(\Phi_{SN})$ by the definition of Cl. □

Consistency of the characteristic formula φ_{SN}

LEMMA 6. *For all $SN \in \mathcal{SN}$, φ_{SN} is **KD45**-consistent.*

PROOF. No negated connection or action predicates are added to Φ_{SN}. Therefore, no inconsistency can be derived from $\{c(i,j) \mid (i,j) \in C_c, c \in \mathcal{C}\} \cup \{a(i,j) \mid (i,j) \in A_a, a \in \Sigma\} \cup \{p(\vec{t}) \mid p(\vec{t}) \in KB_e\}$. Since **KD45** does not include the axiom $K_i\varphi \implies \varphi$ the knowledge of the agents will not make Φ inconsistent. Moreover, the knowledge bases of the agents are consistent by definition, in particular we assume that agents can derive new knowledge according the axiomatisation **S5**. Only what the agents know is included in Φ_{SN}, i.e. Φ_{SN} does not contain any formula of the form $\neg K_i\psi$, therefore no contradiction can be derived. Given the above it follows that φ is consistent. □

D. PRESERVATION OF SATISFIABILITY OF AN SNM CONSTRUCTED FROM A CANONICAL KRIPKE MODEL

We prove that all formulae that hold in the state of the canonical Kripke model corresponding to the characteristic formula are satisfiable in the original SNM. Here, we also provide all the formal definitions required for the proof.

DEFINITION 14 (MARKED CHARACTERISTIC SET). *The* marked characteristic set *of an SNM SN, denoted as Φ^m_{SN}, is constructed as* *follows:*

$$\begin{aligned}\Phi^m_{SN} = \{pr\_p(\overrightarrow{t}) \mid p(\overrightarrow{t}) \in KB_e\} \cup \\ \{co\_c(i,j) \mid (i,j) \in C_c, c \in \mathcal{C}\} \cup \\ \{ac\_a(i,j) \mid (i,j) \in A_a, a \in \Sigma\} \cup \\ \{K_i\varphi \mid \varphi \in KB_i\}\end{aligned}$$

DEFINITION 15 (MARKED KRIPKE TRANSFORMATION FUNCTION). *We define the* marked Kripke transformation function*, denoted $\mathcal{KT}^m$, as $\mathcal{KT}$ in Def. 13 except for step 1), where $\mathcal{KT}^m$ constructs a marked characteristic set (as defined in Def. 14).*

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

PROOF. As we mentioned in Theorem 3, Fagin *et al.* show that all formula φ is satisfiable in a state s_Φ iff $\varphi \in \mathcal{MC}(\Phi)$. Therefore, we can reduce the proof of this theorem to showing that if $\varphi \in \mathcal{MC}(\Phi^m_{SN})$ then $SN \models \varphi$. The proof is split in two cases: 1) $\varphi \in \Phi^m_{SN}$ and 2) $\varphi \in \mathcal{MC}(\Phi^m_{SN}) \setminus \Phi^m_{SN}$.

Case $\varphi \in \Phi^m_{SN}$. We split the proof in the three possible formulas that can be included in Φ^m_{SN} by the definition of 15:

- Case $\varphi = pr\_p(\overrightarrow{t})$. Let $pr\_p(\overrightarrow{t}) \in \Phi^m_{SN}$. By Def. 15 we derive that $p(\overrightarrow{t}) \in KB_e$ and by $\models$ we conclude that $SN \models p(\overrightarrow{t})$ holds.
- Case $\varphi = ac\_a(i,j)$ or $\varphi = a(i,j)$. Let $ac\_a(i,j) \in \Phi^m_{SN}$. By Def. 15 we derive that $p(\overrightarrow{t}) \in A_a$ and by $\models$ we conclude that $SN \models a(i,j)$ holds. The same reasoning holds for $co\_c(i,j)$.
- Case $\varphi = K_i\psi$. Let $K_i\psi \in \Phi^m_{SN}$. By Def. 15 we derive that $\psi \in KB_i$ and by $\models$ we conclude that $SN \models K_i\psi$ holds.

Case $\varphi \in \mathcal{MC}(\Phi^m_{SN}) \setminus \Phi^m_{SN}$. If a formula φ is in $\varphi \in \mathcal{MC}(\Phi^m_{SN}) \setminus \Phi^m_{SN}$ it means that it is has been derived from the explicit knowledge and predicates present in Φ^m_{SN}. As we have shown in Section 4, the axiomatisation **KD45** is sound. Therefore the same derivations can be performed in SN. □

In general, all elements in the a marked characteristic set can be uniquely identified in an SNM, thus it is possible transform a marked characteristic set to the corresponding SNM.

DEFINITION 16 (SNM CONSTRUCTION). *Let $\mathcal{UMS}$ be the the universe of all possible marked characteristic sets. We define the* SNM construction *function, $\mathcal{SC} : \mathcal{UMS} \rightarrow \mathcal{SN}$ as follows: The resulting SNM for Φ^m_{SN}:*

$$SN = \langle Ag, \mathcal{A}, KB, \pi \rangle$$

where

- $Ag = \{i \mid K_i\varphi \in Sub(\varphi^m_{SN})\}$
- *$\mathcal{A}$ contains all function symbols, constant symbols and relation symbols (without the marking) present in the formulae of Φ^m_{SN}.*
  - $KB_e = \{p(\overrightarrow{t}) \mid pr\_p(\overrightarrow{t}) \in \Phi^m_{SN}\}$
  - *for all $c \in \mathcal{C}$, $C_c = \{(i,j) \mid co\_c(i,j) \in \Phi^m_{SN}\}$*
  - *for all $a \in \Sigma$, $A_a = \{(i,j) \mid ac\_a(i,j) \in \Phi^m_{SN}\}$*
- $KB = \{KB_i = \{\varphi \mid K_i\varphi \in \Phi^m_{SN}\}\}_{i \in Ag}$
- $\pi = \emptyset$[5]

E. COMPARISON OF THE COMPLEXITY OF THE SATISFIABILITY PROBLEM IN SNMS AND KRIPKE MODELS

LEMMA 7. *Given $SN \in \mathcal{SN}$ and a formula φ the following holds*

$$O(\sum_{K_i\varphi_i \in outerK(\varphi)} (2^{|\varphi_{KB_i}|} \times |\varphi_i|)) < O(2^{|\varphi_{SN}|} \times |\varphi^K|).$$

PROOF. The formula φ^K is a conjunction of formulae starting with a K_i modality. More specifically, φ^K has the following shape

$$\varphi^K = K_1\varphi^1_1 \wedge K_i\varphi^1_2 \wedge \ldots K_2\varphi^2_1 \wedge K_2\varphi^2_2 \wedge \ldots \wedge K_m\varphi^m_n$$

where $m \in Ag$ and $n \in \mathbb{N}$. Since $K_i\varphi \wedge K_i\psi \implies K_i\varphi \wedge \psi$ holds for any φ and ψ, φ^K can always be rewritten as

$$\varphi^K = K_1\varphi_1 \wedge \ldots \wedge K_m\varphi_m$$

where $\varphi_1 = \varphi^1_1 \wedge \varphi^1_2 \wedge \ldots$ and $\varphi_m = \varphi^m_1 \wedge \varphi^m_2 \wedge \ldots$. Since all φ_m are subformulae of φ^K (without their corresponding K_i modality) the sum of their lenghts will always be strictly smaller, i.e.,

$$|\varphi_1| + \ldots + |\varphi_i| < |\varphi^K|$$

Given the above, the following trivially holds

$$O(|\varphi_1| + \ldots + |\varphi_i|) < O(|\varphi^K|)$$

Multiplying by a constant c in both sides of the inequality does not affect the result

$$O(c \times (|\varphi_1| + \ldots + |\varphi_i|)) < O(c \times |\varphi^K|)$$

By replacing c with $2^{|\varphi_{SN}|}$ in the previous statement we obtain the following

$$O(2^{|\varphi_{SN}|} \times (|\varphi_1| + \ldots + |\varphi_i|)) < O(2^{|\varphi_{SN}|} \times |\varphi^K|)$$

Since multiplication distributes over addition the following holds

$$O(2^{|\varphi_{SN}|} \times |\varphi_1| + \ldots + 2^{|\varphi_{SN}|} \times |\varphi_i|) < O(2^{|\varphi_{SN}|} \times |\varphi^K|) \quad (1)$$

As we describe in Section 6 φ_{KB_i} represents the conjunction of all formulae in the knowledge base of agent i. Besides, the characteristic formula φ_{SN} is a conjunction of the formulae of the knowledge bases of all agents plus the normal predicates. Therefore for any $i \in Ag$ it holds that $2^{|\varphi_{KB_i}|} < 2^{|\varphi_{SN}|}$. From this fact and (1) we derive the following

$$O(2^{|\varphi_{KB_1}|} \times |\varphi_1| + \ldots + 2^{|\varphi_{KB_i}|} \times |\varphi_i|) < O(2^{|\varphi_{SN}|} \times |\varphi_1| + \ldots + 2^{|\varphi_{SN}|} \times |\varphi_i|) \quad (2)$$

Finally, by transitivity of $<$ in (1) and (2) we conclude that

$$O(2^{|\varphi_{KB_1}|} \times |\varphi_1| + \ldots + 2^{|\varphi_{KB_i}|} \times |\varphi_i|) < O(2^{|\varphi_{SN}|} \times |\varphi^K|)$$

as required. □ □

[5] π is empty since Kripke models do not contain information about the privacy policies of the agents.