Journal of Applied Logic 13 (2015) 48–77

Contents lists available at ScienceDirect

Journal of Applied Logic

www.elsevier.com/locate/jal

# A logic for diffusion in social networks

Zoé Christoff [a,*], Jens Ulrik Hansen [b,*]

[a] Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam, Netherlands
[b] Department of Philosophy, Lund University, Sweden

ARTICLE INFO

*Article history:*
Received 19 May 2014
Accepted 5 September 2014
Available online 27 November 2014

*Keywords:*
Modal logic
Hybrid logic
Social networks analysis
Diffusion in social networks

ABSTRACT

This paper introduces a general logical framework for reasoning about diffusion processes within social networks. The new "Logic for Diffusion in Social Networks" is a dynamic extension of standard hybrid logic, allowing to model complex phenomena involving several properties of agents. We provide a complete axiomatization and a terminating and complete tableau system for this logic and show how to apply the framework to diffusion phenomena documented in social networks analysis.


## 1. Introduction

Social networks are groups of agents structured by some social relationship/links such as family ties, being colleagues, or "following" on social media sites – in other words directed or undirected graphs. In the last decade, the study of social networks, and networks in general, has seen a rapid increase (classical textbooks include [33,26,18]). A variety of aspects of networks have been studied: how they emerge, how they change, what their structural properties are, what social roles they play, etc. This paper focuses on dynamic processes occurring within social networks, such as diffusion of information, viruses, trends, opinions, or behaviors, for instance. What typically characterizes such processes is that their dynamics are local: whether an agent adopts a behavior/opinion/disease/product/trend depends on whether the agents linked to him within the social network have adopted it already. Since social networks are graphs and since the dynamics depend on *local* properties, it seems natural to develop a dynamic *modal* logic to reason about such phenomena in social networks. This is exactly what this paper does.

### 1.1. Diffusion phenomena

To clarify the type of processes our framework is designed for, let us start by considering an example: the diffusion of a disease within a population. Assume that each agent of the population is in either of

* Corresponding authors.
*E-mail addresses:* zoe.christoff@gmail.com (Z. Christoff), jensuhansen@gmail.com (J.U. Hansen).

http://dx.doi.org/10.1016/j.jal.2014.11.011
1570-8683/

two states: *infected* with the disease or *susceptible* to it. This type of models is commonly called "the SI Model" in the social networks literature [33]. Moreover, assume that the disease can only be contracted by being physically in contact with an infected agent. Consider the network consisting of agents in a given population, where two agents are linked if they are in contact with each other. (If an agent $a$ is linked to an agent $b$, we will also call $b$ a "neighbor" of $a$ or a "friend" of $a$.)

Consider now how such an infection spreads through a (finite) population. This depends on how contagious the disease is. Assume that each agent linked to an infected agent in the network will get infected too at the next moment. This means that if some agent $i$ is infected to start with, all agents directly linked to him will be infected at the next moment, and then all agents linked to the agents linked to him, and so on. Finally, according to this rule of contagion, all agents in (the connected component of) the population will contract the disease after some time. However, note that the social network structure constrains how fast such a disease spreads and what measures would be needed to contain it – the shortest network-path from an agent $a$ to the initially infected agent $i$ determines how long it takes before agent $a$ will get infected. In this example, the dynamics is essentially captured by the following local diffusion rule: If any of your neighbors is infected, become infected yourself at the next moment.

The long term dynamics of such contagion phenomena can also be investigated. Assuming that once an agent gets infected she will stay infected forever, each finite connected component of the network will reach a *stable* state where everybody is infected. However, one could very well imagine a different diffusion rule: after being infected, an agent immediately recovers and becomes susceptible again at the next moment. According to this new dynamic rule, agents might keep alternating forever between being infected and being susceptible and the network might never reach a stable state.

This simple "SI" example can be enriched in several ways. First, the health status of an agent could take more than two values. In the so-called "SIR Model" (*susceptible, infected, recovered*, see for instance [33]), agents can become "recovered" after being infected, which might mean that they have become immune to the disease or that they will move back to being susceptible.

In the above, only the current health status of the agents matters to determine their future health status. A second way to enrich such diffusion models is to take into account other features of agents which might interfere with the diffusion of the infection. For instance, imagine a genetic type such that agents of this type are immune to the disease or stay infected for longer. In this case, the epidemic behavior reveal more complexity, and the diffusion rule needs to combine several properties of agents to take into account all factors.

Two aspects of the above examples will be particularly relevant to this paper. First, agents have certain *properties* such as health status, genetic type, age, gender, hair color, etc. For each agent all these properties are instantiated by particular *features* (or *values*), such as infected, is of the immune genetic type, 34 years old, female, redhead, etc. The features of some of these properties are spreading within the network (in our simple examples, only the health status features are). For each property the associated possible features will come from some fixed set of values, such as: the three possible health states, numbers 1 to 130, male or female, a set of possible hair colors, etc. This assignment of one value to each property will be captured by a particular kind of atomic propositions in the logic developed in the next section.

The second thing to remark is that the dynamics are defined in a purely *local* way. In the above, an agent changes her health status from being susceptible to being infected if at least one of her neighbors is infected. Other kinds of local dynamics could be considered: for instance, dying your hair red if all of your friends have red hair or if at least one of your friends has a friend who has red hair. This type of local conditions is ideally described by formulas of a modal language. Thus, using an extension of basic modal logic will provide a natural way of defining a large variety of dynamic processes on social networks.

### 1.2. Contributions of the paper

Traditionally, network science and logic have co-existed as separated fields. The type of contagion just mentioned, as well as numerous other diffusion phenomena, have been widely studied in the network science literature (again see for instance [33,26,18]). In parallel, the dynamics of interacting agents have also been thoroughly addressed within the field of logic [45,6,46]. However, only a fraction of this line of work in logic has taken the social network structure into account.[1] Nevertheless, the idea of introducing notions from network science into logic is progressively emerging in very recent work [44,49,41,43,29,16,15,24]. This paper attempts to continue building the bridge between logic and network analysis.

The first novelty of the paper is the way preexisting modal logics are extended with a dynamic part allowing to model complex locally governed dynamic phenomena within social network structures. In this respect, we build on some of our earlier work: the idea of equipping agents with different properties with different values was developed in [16], where an early version of the framework introduced here was used to model specifically the phenomenon of pluralistic ignorance (see Section 5.1). The idea of using modal formulas to define dynamics was also introduced in [16] as a way of generalizing the simple belief dynamics of [29].[2]

Secondly, unlike the more philosophical case study provided by [16], this paper contains a general and technical presentation of the logic. It provides a complete axiomatization as well as a sound, complete and terminating tableau system for the new Logic for Diffusion in Social Networks. Our logic extends standard hybrid logic [2], which offers relevant additional expressive powers compared to basic modal logic: by naming agents in the network, additional structural properties of frames can be captured, as already noted by [8,9,44]. The axiomatization of our underlying static logic is very similar to the axiomatizations of [11,2] and the additional axiomatization of our full dynamic logic borrows the reduction technique from [6,46]. The tableau system for the static logic is adapted from [13], while the tableau system for our dynamic logic is inspired by the technique of [23]. Our proofs of termination and completeness of the tableau system for our full dynamic logic rely heavily on the corresponding proofs for the static logic, as provided by [13].

Thirdly, the paper exemplifies how the new general framework applies to particular social phenomena which have been well documented within the literature, such as the diffusion of micro-loans in villages, and gives a general overview of existing work combining network structures and logic.

The paper is structured as follows: Section 2 defines the Logic for Diffusion in Social Networks. The static part of the logic is introduced first and *dynamic transformations* are then added to it to obtain the full Logic for Diffusion in Social Networks. Section 3 provides a complete axiomatization of the logic, while Section 4 provides a complete and terminating tableau system for it. Section 5 exemplifies how the logic allows reasoning about real-life network behaviors. Finally, Section 6 discusses related work, possible extensions of the framework, and future research.

## 2. Logic for diffusion in social networks

In this section, we introduce a hybrid logical framework to reason about the change of distribution of features among agents within social networks. We start with the static part of the logic, which we call *Logic for Social Networks*, to model the situation of agents in the network at a given moment. We then move on to the full dynamic *Logic for Diffusion in Social Networks* to represent the evolution of such situations.

[1] See Section 6 for exceptions.

[2] The way of defining the dynamics is also partly inspired by the events modalities of Dynamic Epistemic Logic [6,46].

### 2.1. Logic for social networks

As in [44], our framework includes standard tools from hybrid logic [2] to be able to talk about the network structure. Hence, following [44], our formulas will have an indexical reading. The main novelty in our static logic is the format of the atomic propositions used to talk about properties of the agents. Recall that we view agents as having certain features that are instantiations of some fixed properties under consideration. Instead of a set of standard propositional variables, we use equational statements to talk about features of agents. We assume that each agent has $n$ different relevant properties, to each of which is assigned one value from a finite set. To avoid any later confusion, let us first remind our reader of the vocabulary we will be using: we use the term "property" to refer to for instance age, gender, health status, etc., and the term "feature" to refer to the value assigned to such a property, for instance 34 years old, infected, redhead, etc. In this sense, a property is a "feature variable" and a feature is a value taken by this variable.

More formally, we fix a finite set of *feature variables* $\{V_1, V_2, ..., V_n\}$ representing $n$ different properties of agents, where each variable $V_l$ is associated with a given finite *value set* $R_l$. The atomic propositions (or feature propositions) of our language will then be defined in the following way:

**Definition 1** *(Feature propositions).* A *feature proposition* is of the form

$$V_l = r,$$

for some $l \in \{1, \dots, n\}$ and some $r \in R_l$. The set of all feature propositions (for fixed sets of variables and values) will be denoted $\mathsf{FP}$.

The intuition is that the proposition $V_l = r$ is true of an agent if and only if the agent possesses feature $r$ of property $V_l$. For instance, assuming that we have two properties $V_g$ for gender and $V_h$ for health status, we could write $V_g = f$ to express that an agent is female and $V_h = i$ to express that an agent is infected.[3]

In addition to the finite set of feature propositions ($\mathsf{FP}$), we will assume a countable infinite set of nominals ($\mathsf{NOM}$) used as names for agents in networks, just as nominals are used to refer to possible states in traditional hybrid logic [2]. We can now give the syntax for the static social networks language:

**Definition 2** *(Syntax for social networks language $\mathcal{L}_{\mathcal{SN}}$).* The syntax of the social networks language, denoted $\mathcal{L}_{\mathcal{SN}}$, is given by:

$$\varphi ::= V_l = r \mid i \mid \neg\varphi \mid \varphi \wedge \varphi \mid F\varphi \mid U\varphi \mid @_i\varphi,$$

where $V_l = r \in \mathsf{FP}$ and $i \in \mathsf{NOM}$.

We will use the standard abbreviations for $\vee$, $\rightarrow$, and $\leftrightarrow$ and denote the dual operator of $F$ by $\langle F\rangle$ and the dual of $U$ by $\langle U\rangle$. Moreover, we define $\bigwedge_{i=1}^{n} \varphi_i := \bigwedge_{\varphi\in\{\varphi_1,\dots,\varphi_n\}} \varphi := (...((\varphi_1 \wedge \varphi_2) \wedge \varphi_3) \wedge ...) \wedge \varphi_n$ and $\bigvee_{\varphi\in\{\varphi_1,\dots,\varphi_2\}} := (...((\varphi_1 \vee \varphi_2) \vee \varphi_3) \vee ...) \vee \varphi_n$. The intuitive meaning of a formula $F\varphi$ is that "$\varphi$ is true of all my network-neighbors" (or "friends", to stick to the terminology of [29] and the intuitive meaning of a formula $@_i\varphi$ is that "$\varphi$ is true of the agent named $i$" – note the indexical reading of formulas here! The $U$-operator is the global modality quantifying over all agents in the network and $U\varphi$ is read as "$\varphi$ is true of all agents in the network".

[3] Note that feature propositions can be viewed as a generalization of classical propositional variables. Given a classical propositional variable $P$ one can add a variable $V_P$ and let $R_P = \{1, 0\}$. Then $V_P = 1$ will represent that $P$ is true and $V_P = 0$ will represent that $P$ is false (i.e. $\neg P$).

As previously mentioned, we have assumed that a fixed set of feature propositions $\mathsf{FP}$ is given. This assumption will be made throughout the rest of the paper unless otherwise specified. Whenever we need to be explicit about the set of feature propositions to which our language is relative, we will use the notation $\mathcal{L}_{\mathcal{SN}}(\mathsf{FP})$.

Before defining the semantics, let us first introduce the notion of assignment and define our models. Intuitively, an assignment will assign specific values to the set of variables, hence determining the features of a given agent:

**Definition 3** *(Assignment/full assignment).* An *assignment* (or *partial assignment*) is a partial function $s$ from $\{1, ..., n\}$ to $\bigcup_{l=1}^{n} R_l$ such that $s(l) \in R_l$ for all $1 \leq l \leq n$ where $s$ is defined. The set of all assignments is denoted by $\mathcal{V}$. For a given assignment $s$, the *domain* of $s$ is denoted by $dom(s)$. A *full assignment* is an assignment $s$ such that $dom(s) = \{1, ..., n\}$. The set of all full assignments is denoted $\mathcal{V}^{full}$.

The idea is that an assignment $s$ assigns a feature $s(l) \in R_l$ to the feature variable $V_l$, for each $l \in dom(s)$. Thus, a *full* assignment $s$ assigns a feature $s(l)$ to *every* feature variable $V_l$ ($l \in \{1, ..., n\}$).[4]

**Definition 4** *(Network model).* A *network model* is a tuple $\mathcal{M} = (A, \asymp, g, \nu)$, where: $A$ is a non-empty set of agents, $\asymp$ is a binary relation on $A$ representing the network structure, $g : \mathsf{NOM} \to A$ is a function assigning an agent to each nominal, and $\nu : A \to \mathcal{V}^{full}$ is a valuation assigning a full assignment $\nu(a)$ to each agent $a \in A$, i.e. a complete specification of the features of each agent in the network. The pair $(A, \asymp)$ will be referred to as a *frame* and a model built on a frame $(A, \asymp)$ is simply a model obtained by adding a $g$ and a $\nu$ to the frame.

For instance, if we have two properties or "feature variables", health status and gender, a full assignment assigns one value for each variable to each agent in the network. In other words, no property of any agent is left undefined in a model. We can move on to the semantics for the language $\mathcal{L}_{\mathcal{SN}}$:

**Definition 5** *(Semantics of $\mathcal{L}_{\mathcal{SN}}$).* Given a $\mathcal{M} = (A, \asymp, g, \nu)$, an $a \in A$ and a formula $\varphi \in \mathcal{L}_{\mathcal{SN}}$, we define the truth of $\varphi$ at $a$ in $\mathcal{M}$ inductively by:

$$
\begin{array}{rll}
\mathcal{M}, a \models V_l = r & \text{iff} & \nu(a)(l) = r \\
\mathcal{M}, a \models i & \text{iff} & g(i) = a \\
\mathcal{M}, a \models \neg\varphi & \text{iff} & \text{it is not the case that } \mathcal{M}, a \models \varphi \\
\mathcal{M}, a \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, a \models \varphi \text{ and } \mathcal{M}, a \models \psi \\
\mathcal{M}, a \models U\varphi & \text{iff} & \text{for all } b \in A; \mathcal{M}, b \models \varphi \\
\mathcal{M}, a \models F\varphi & \text{iff} & \text{for all } b \in A; a \asymp b \text{ implies } \mathcal{M}, b \models \varphi \\
\mathcal{M}, a \models @_i\varphi & \text{iff} & \mathcal{M}, g(i) \models \varphi
\end{array}
$$

We say that a formula $\varphi$ is *satisfiable* if there is a model $\mathcal{M} = (A, \asymp, g, \nu)$ and an agent $a \in A$ such that $\mathcal{M}, a \models \varphi$ (and unsatisfiable otherwise). If this is the case, we will also simply say that $a$ satisfies $\varphi$ (taking $\mathcal{M}$ to be given). Two formulas $\varphi$ and $\psi$ are said to be *pairwise unsatisfiable* if $\varphi \wedge \psi$ is unsatisfiable. Given a model $\mathcal{M} = (A, \asymp, g, \nu)$ and a formula $\varphi$ we write $\mathcal{M} \models \varphi$ if $\mathcal{M}, a \models \varphi$ for all $a \in A$. A formula $\varphi$ is said to be *valid with respect to a class of frames* if $\mathcal{M} \models \varphi$ for all models $\mathcal{M}$ built on some frame from the class.

[4] The reader might notice that we only need *full* assignments to define models. However, partial assignments will simplify things when we will define our dynamic Logic for Diffusion in Social Networks in Section 2.2.

A formula is said to be just *valid* if it is valid with respect to the class of all frames. The logic consisting of the set of all valid formulas will be denoted LSN and referred to as the logic for social networks.

While choosing a *modal* language seems natural to describe network structures, the reader might wonder why we choose a *hybrid* one. Ref. [8] has shown that some global properties of graphs standardly discussed in graph theory are neither definable in basic modal language (even if one adds a transitive closure modal operator to the language) nor in any bisimulation invariant extension of it, such as modal $\mu$-calculus: connectivity, acyclicity, and Hamiltonian property (i.e., whether there is a cycle passing through each vertex of a graph exactly once), for instance. Add nominals and $@_i$ normal modal operators and all those properties become definable, as [8] shows. While our language does not include the transitive closure operator used in [8] and therefore cannot express connectivity and acyclicity with the same succinctness,[5] it can express the Hamiltonian property in the exact same way introduced in [8].

We leave the full expressivity comparison between the present framework and others for future research, but give four examples of global properties of networks expressible in our language: irreflexivity (no agent is linked to itself), symmetry (if a first agent is linked to a second one, then the second one is also linked to the first one), full connectivity (all different agents are linked to each other), $n$-connectivity (there is a path of length at most $n$ between any two pair of different agents):

A frame $(A, \asymp)$ is

*irreflexive* iff $(A, \asymp) \models @_i \neg \langle F \rangle i$,
*symmetry* iff $(A, \asymp) \models @_i \langle F \rangle j \rightarrow @_j \langle F \rangle i$,
*fully connected* iff $(A, \asymp) \models @_i (\neg j \rightarrow \langle F \rangle j)$,
*$n$-connected* $(n \in \mathbb{N} \geq 1)$ iff $(A, \asymp) \models @_i (\neg j \rightarrow (\langle F \rangle j \vee \langle F \rangle^2 j \vee \ldots . \langle F \rangle^n j))$, where $\langle F \rangle^k$ is given by:
$\langle F \rangle^1 := \langle F \rangle$
$\langle F \rangle^{k+1} := \langle F \rangle \langle F \rangle^k$.

Those global connectivity properties will impact the diffusion phenomena we are interested in. For instance, recall the initial epidemic "SI model" example from Section 1: if a network is $n$-connected, all agents will become infected after a period of length at most $n$. Hence, the hybrid static framework introduced so far is rather promising: it is expressive enough to describe the distribution of features among agents in the network, as well as some network properties which are relevant to how those features will be redistributed in the future. In the next subsection, we introduce the tools to talk precisely about *changes* or *transformations* of such network situations.

### 2.2. Dynamic transformations

We now extend our logic to deal with the dynamics of networks. We make two important design choices. First, we will be concerned exclusively with one particular type of change: the change of distribution of features of agents within a social network structure. This means that we assume that agents do not change names and that the network structure is fixed. Second, we take a very general point of view: our agents are essentially just bundles of features with stable names, and the question of *how* such features should change is so open-ended that we consider that the safest option is to offer a framework which is general enough to allow for any such type of change, as long as it is locally definable in terms of our language. Consequently, our setting can be refined in many ways to accommodate different types of applications and represent their corresponding dynamics. Our framework allows to "plug-in": 1) how many properties of agents are relevant, 2) how many values each of these properties can take and 3) according to which rules such static models

[5] Section 6.3 discusses extensions of our framework with transitive closure operators (of both the static and dynamic modalities).

should be updated, i.e., how those features should be redistributed on the network. In other words, we are abstracting as much as possible from particular diffusion examples given by the networks analysis literature by building a framework which can deal with most of them.

What we want is a way to obtain a new model from a given model through some transformation. In this respect, our dynamic modalities will be comparable to the event models modalities of Dynamic Epistemic Logic [6,46]. However, instead of event models, we will talk about *dynamic transformations*.

**Definition 6** *(Dynamic transformations).* A *dynamic transformation* is a pair $\mathcal{D} = (\Phi, \mathsf{post})$ consisting of a non-empty finite set $\Phi$ of pairwise unsatisfiable formulas (from the language $\mathcal{L}_{\mathcal{DSN}}$ to be defined in Definition 7)[6] and a post-condition function $\mathsf{post} : \Phi \to \mathcal{V}$. The set $\Phi$ will be referred to as "preconditions", and given a precondition $\varphi \in \Phi$, we will call the assignment $\mathsf{post}(\varphi) \in \mathcal{V}$ the "post-condition" of $\varphi$.

Note that the post-conditions are partial assignments and not full assignments. The intuition behind this definition is that if an agent satisfies a $\varphi \in \Phi$ (in which case, $\varphi$ is necessarily unique), then after the dynamic transformation $\mathcal{D}$, $a$ changes her features as specified by $\mathsf{post}(\varphi)$. As $\mathsf{post}(\varphi)$ is a partial assignment $a$ does not change all her features, only the ones in $dom(\mathsf{post}(\varphi))$.

On the syntactic level, we add formulas of the form $[\mathcal{D}]\varphi$ for a given dynamic transformation $\mathcal{D}$. In the following we consider a fixed set of dynamic transformations to be given and denote it by $\mathsf{DT}$. Here is the syntax of our full dynamic language:

**Definition 7** *(Syntax of language for diffusion in social networks $\mathcal{L}_{\mathcal{DSN}}$).* The syntax of the language for diffusion in social networks, denoted $\mathcal{L}_{\mathcal{DSN}}$, is given by:

$$\varphi ::= V_l = r \mid i \mid \neg\varphi \mid \varphi \wedge \varphi \mid F\varphi \mid U\varphi \mid @_i\varphi \mid [\mathcal{D}]\varphi,$$

where $V_l = r \in \mathsf{FP}$, $i \in \mathsf{NOM}$, and $\mathcal{D} \in \mathsf{DT}$.

As for the static language $\mathcal{L}_{\mathcal{SN}}$, whenever needed, to make explicit which set of dynamic transformations $\mathsf{DT}$ (and feature propositions $\mathsf{FP}$) a language is built upon, we use the notation $\mathcal{L}_{\mathcal{DSN}}(\mathsf{DT})$ ($\mathcal{L}_{\mathcal{DSN}}(\mathsf{FP}, \mathsf{DT})$).

The satisfaction of formulas involving dynamic modalities relies on transforming the model at hand. This is captured by the following definition:

**Definition 8** *(Transformation updates).* Given a model $\mathcal{M} = (A, \asymp, g, \nu)$ and a dynamic transformation $\mathcal{D} = (\Phi, \mathsf{post})$, the *updated* model under the transformation $\mathcal{D}$ is $\mathcal{M}^{\mathcal{D}} = (A, \asymp, g, \nu')$, where $\nu'$ is defined by:

$$\nu'(a)(l) = \begin{cases} \mathsf{post}(\varphi)(l) & \text{if there is a } \varphi \in \Phi \text{ such that } \mathcal{M}, a \models \varphi \\ & \text{and } l \in dom(\mathsf{post}(\varphi)) \\ \nu(a)(l) & \text{otherwise} \end{cases} \tag{1}$$

for all $a \in A$ and all $l \in \{1, ..., n\}$.

As previously mentioned, the intuition is that if an agent satisfies a $\varphi \in \Phi$ then, after the dynamic transformation $\mathcal{D}$, she changes her features as specified by $\mathsf{post}(\varphi)$. More formally, assume that an agent $a$

[6] We have to be a little careful here. To avoid circular definitions we cannot allow the dynamic transformation $\mathcal{D} = (\Phi, \mathsf{post})$ to have precondition formulas in $\Phi$ involving $\mathcal{D}$ itself. Nevertheless, we can allow formulas of $\mathcal{L}_{\mathcal{DSN}}$ in $\Phi$ constructed on an "earlier stage" in a simultaneous inductive definition of dynamic transformations and the language $\mathcal{L}_{\mathcal{DSN}}$. In other words, one should view Definition 6 and Definition 7 as one simultaneous recursive definition. The issue is similar to the issue of defining the full language of Dynamic Epistemic Logic [46, Ch. 6].

satisfies $\varphi$ and consider the variable $V_l$. If $l \notin dom(\mathsf{post}(\varphi))$, then $V_l = r$ will be true of $a$ after $\mathcal{D}$ if, and only, if $V_l = r$ was true of $a$ before $\mathcal{D}$. On the other hand, if $l \in dom(\mathsf{post}(\varphi))$, then $V_l = r$ will be true of $a$ after $\mathcal{D}$ if, and only, if $\mathsf{post}(\varphi)(l) = r$. Note that the "otherwise" case in (1) takes care of two situations, namely the situation where there are no formulas in $\Phi$ true of the agent $a$, and the situation where there might be a formula $\varphi \in \Phi$ true of $a$, but the feature in question, $l$, is not in the domain of $\mathsf{post}(\varphi)$.

The semantics of the dynamic language can now be given:

**Definition 9** *(Semantics of $\mathcal{L}_{\mathcal{DSN}}$).* Given a $\mathcal{M} = (A, \asymp, g, \nu)$, an $a \in A$ and a formula $\varphi \in \mathcal{L}_{\mathcal{DSN}}$, we define the truth of $\varphi$ at $a$ in $\mathcal{M}$ inductively as in Definition 5 with the additional clause:

$$\mathcal{M}, a \models [\mathcal{D}]\varphi \quad \text{iff} \quad \mathcal{M}^{\mathcal{D}}, a \models \varphi.$$

Satisfiability, validity, and pairwise unsatisfiability are generalized in the obvious way from Definition 5. The logic consisting of the set of all valid $\mathcal{L}_{\mathcal{DSN}}$-formulas will be denoted LDSN and referred to as Logic for Diffusion in Social Networks.

Before moving on, let us consider the small example from the introduction concerning diffusion of a disease again. Here we might have two variables $V_{HS}$ and $V_{GI}$ keeping track of the health status of the agents and whether they are genetically immune, i.e. $R_{HS} = \{susceptible, infected, recovered\}$ and $R_{GI} = \{yes, no\}$. Thus, $V_{HS} = susceptible \wedge V_{GI} = no$ is true of an agent if she is susceptible to the disease and she is not genetically immune. We could then specify the following dynamic transformation $\mathcal{D} = (\Phi, \mathsf{post})$, for instance:

| $\Phi$: | post: |
|---|---|
| $V_{HS} = susceptible \wedge V_{GI} = no \wedge \langle F \rangle V_{HS} = infected$ | $\mathsf{post}(HS) = infected$ |
| $V_{GI} = yes$ | $\mathsf{post}(HS) = recovered$ |
| $V_{HS} = infected$ | $\mathsf{post}(HS) = recovered$ |

This dynamic transformation represents the fact that a non-genetically immune susceptible agent becomes infected if at least one of her neighbors is infected, while a genetically immune agent is immediately recovered and does not get infected ever. Moreover, after being infected an agent moves to being recovered. As there is no specification of how agents would move from being recovered to being susceptible (or infected), recovered agents become immune to the disease too. This example is minimal in the sense that it involves only two properties of agents and assumes that one property is spreading (health status) while the other is not (genetic immunity status). However, our framework can go way beyond this simple example as it allows for combining several properties, spreading or not. More complex examples of applications of LDSN can be found in Section 5.

It follows from Definition 8 that, for every network model $\mathcal{M}$ and every dynamic transformation $\mathcal{D}$, the updated network model $\mathcal{M}^{\mathcal{D}}$ always exists. Moreover, no agent from $A$ is deleted when moving to the new model $\mathcal{M}^{\mathcal{D}}$. Thus for every pair $(\mathcal{M}, a)$ of a network model $\mathcal{M}$ and an agent $a$ of $\mathcal{M}$, and for every dynamic transformation $\mathcal{D}$, the pair $(\mathcal{M}^{\mathcal{D}}, a)$ exists. Hence, contrary to public announcement logic or traditional dynamic epistemic logic, a formula can be evaluated in a state $(\mathcal{M}, a)$ exactly when it can be evaluated in a state $(\mathcal{M}^{\mathcal{D}}, a)$. For this reason, there is no need to give the semantics of the dynamic modality $[\mathcal{D}]$ in term of a conditional clause. Moreover, it implies that dynamic transformations are "functional", in the sense that each dynamic transformation $\mathcal{D}$ behaves as a function on the class of pointed network models $(\mathcal{M}, a)$, in accordance with most of the examples in the network analysis literature. This is reflected in the logic by the fact that all dynamic transformations are their own duals, i.e.

$$[\mathcal{D}]\varphi \leftrightarrow \neg[\mathcal{D}]\neg\varphi,$$

is a validity for all dynamic transformations $\mathcal{D}$ and all formulas $\varphi$. For these reasons, we can always define the sequential application of the same dynamic transformation $\mathcal{D}$ in a straightforward way as follows:

**Definition 10** *($\mathcal{M}^{k\mathcal{D}}$).* Given a network model $\mathcal{M}$ and a dynamic transformation $\mathcal{D}$, let $\mathcal{M}^{k\mathcal{D}}$ be defined recursively for all $k \in \mathbb{N}_0$ by:

$$\mathcal{M}^{0\mathcal{D}} := \mathcal{M}$$
$$\mathcal{M}^{(k+1)\mathcal{D}} := \left(\mathcal{M}^{k\mathcal{D}}\right)^{\mathcal{D}}.$$

Some long-term behaviors of networks can be observed. Given a network model $\mathcal{M}$ and a dynamic transformation $\mathcal{D}$, an interesting question is whether the network stabilizes, that is, whether successive updates by $\mathcal{D}$ will result in a network model which does not change under further update by $\mathcal{D}$, i.e. a fixed-point of the model transformation $\mathcal{D}$.

**Definition 11** *(Stability of a model).* A network model $\mathcal{M}$ is said to be *stable* under a dynamic transformation $\mathcal{D}$, if $\mathcal{M} = \mathcal{M}^{\mathcal{D}}$. $\mathcal{M}$ is said to *stabilize* under the dynamic transformation $\mathcal{D}$, if there is a $k \in \mathbb{N}_0$ such that $\mathcal{M}^{k\mathcal{D}}$ is stable.

Can our logic say something about such limit behaviors of networks? Yes, it can: it can capture the notion of stability. Let us explain how to express in our language that a network is stable.[7] Given a model $\mathcal{M} = (A, \asymp, g, \nu)$, the full assignment $\nu(a)$ completely describes the features of $a$, thus the complete features of $a$ is expressed by:

$$\varphi_{\nu(a)} := \bigwedge_{l=1}^{n} V_l = \nu(a)(l).$$

Moreover, note that the set of all possible full assignments $\mathcal{V}$ is finite. Thus, we can "quantify" over it in our language and express that a network model is stable under $\mathcal{D}$ by[8]:

$$\varphi_{stable(\mathcal{D})} := \bigwedge_{s \in \mathcal{V}^{full}} \left(\varphi_s \to [\mathcal{D}]\varphi_s\right). \tag{2}$$

That this is in fact so follows from the following lemma:

**Lemma 12.** *A network model $\mathcal{M}$ is stable under $\mathcal{D}$ if, and only if,*

$$\mathcal{M} \models \varphi_{stable(\mathcal{D})}.$$

Let us summarize what we have done so far. First, we have defined a static logic to talk about features of agents in a social network. Then, we have defined the set of transformations of the distribution of those features which are locally definable in terms of preconditions and postconditions within our (restricted)

[7] While our language can express stability, it cannot express stabilization. The straight forward way would be to add the Propositional Dynamic Logic (PDL) transitive closure construct $*$ to our modality $[\mathcal{D}]$ [25]. However, it would be interesting to find a formula without the $\langle \mathcal{D}^* \rangle$ operator that defines stabilizing networks, just as done in [49] for the case of a logic of preference change. We leave this for future research. Stabilization and the transitive closure operator $*$ are discussed in more details in the concluding Section 6.

[8] Another way of expressing that a network model is stable would be to follow the line of [29]. If $V_L = r$ is true of some agent and the network is stable, this means that none of the preconditions $\varphi \in \Phi$ of $\mathcal{D}$ for which $\mathsf{post}(\varphi)$ would change the value of $V_l$ can be satisfied at the agent. Then, for every feature we can write the conjunction of the negation of all preconditions that would change this feature. Finally, we can take the disjunction over all possible features and thereby obtain a formula for a network being stable. This, of course, would result in a much more complex formula, however, it would avoid the explicit use of the $[\mathcal{D}]$ modality.

language. Moreover, we have shown that our language can capture some static properties of networks, such as $n$-connectedness, and some dynamic properties of network models such as stability. In a nutshell, we have presented a logic able to describe the type of states of social networks and the type of changes which we wanted to capture. In the next section, we will consider what kind of *reasoning* about those networks is supported by our logic, by giving a complete proof-system for it.

## 3. Axiomatizations

In this section, we will provide sound and complete Hilbert-style proof systems for the logics of Section 2. The axiomatization of the static logic LSN follows that of [11,2] with a few modifications, while the axiomatization of the dynamic logic LDSN expands that of the static logic with "reduction axioms" – a standard technique of Dynamic Epistemic Logic [6,46].

Before giving the Hilbert-style axiomatization of LSN, we recall some standard terminology for Hilbert-style proof systems: A proof of $\varphi$ is a finite sequence of formulas ending with $\varphi$ such that every formula in the sequence is either an axiom or follows from previous formulas in the sequence using one of the proof rules. We denote this by $\vdash \varphi$. We use $\vdash_{\mathsf{S}}$ for provable in the proof-system for LSN and $\vdash_{\mathsf{D}}$ for provable in the proof-system for LDSN. In the following, X will thus stand for either S or D. For a set of formulas $\Gamma$, $\Gamma \vdash_{\mathsf{X}} \varphi$ holds if there are $\psi_1, ..., \psi_n \in \Gamma$ such that $\vdash_{\mathsf{X}} \psi_1 \wedge ... \wedge \psi_n \to \varphi$. Given a set of formulas $\Sigma$, let $\mathsf{X} + \Sigma$ denote the logic obtained by adding all the formulas in $\Sigma$ as axioms. That $\varphi$ is provable in the logic $\mathsf{X} + \Sigma$ will then be denoted by $\vdash_{\mathsf{X}+\Sigma} \varphi$. A set of formulas $\Gamma$ is said to be $\mathsf{X} + \Sigma$-inconsistent if $\Gamma \vdash_{\mathsf{X}+\Sigma} \bot$, and $\mathsf{X} + \Sigma$-consistent otherwise. A formula $\varphi$ is pure if it does not contain any feature propositions. A set of formulas $\Sigma$ is called substitution-closed if it is closed under uniform substitution of nominals by nominals.

### *3.1. Complete axiomatization of* LSN

The Hilbert-style axiomatization of LSN is shown in Fig. 1. As previously mentioned, the axiomatization is fairly standard in the hybrid logic literature except for the axioms Char.Prop.1 and Char.Prop.2. While Char.Prop.1 ensures every variable $V_l$ is assigned at least one value, Char.Prop.2 ensures that no variable $V_l$ is assigned more than one value.

Soundness and completeness of this type of axiomatization are also standard results in the hybrid logic literature (see [11,2]). Thus, we do not include the proofs of these properties and only state the completeness theorem here:

**Theorem 1** *(Completeness of* LSN*). Let $\Sigma$ be a substitution-closed set of pure formulas. Every set of formulas that is* $\mathsf{LSN} + \Sigma$*-consistent is satisfiable in a model whose underlying frame validates all the formulas in $\Sigma$.*

This form of completeness theorem is typical for hybrid logic and highlights one of its advantages. As traditional with completeness proofs, when assuming that a formula $\varphi$ is valid and that it is not provable, the set $\{\neg\varphi\}$ becomes consistent. Then, by the above theorem, we can derive a counter-model to $\varphi$, which yields a contradiction to the assumption that $\varphi$ was not valid – thus $\varphi$ must be provable. The benefit of the hybrid logic is that we can make sure that this counter-model is of a typical kind, namely a model where each world is named by a nominal. This further implies that the underlying frame validates all formulas in $\Sigma$ (as $\Sigma$ is substitution closed). Thus, our counter-model is based on a frame from the class of frames defined by $\Sigma$ and we thereby automatically achieve completeness with respect to the class of frames defined by $\Sigma$. (Going through an extension of the standard translation of modal logic into first-order logic, one can see that pure formulas will always define first-order properties of frames [12].)

The strength of this kind of automatic completeness is easily illustrated by considering a particular class of networks, namely networks where the relation $\asymp$ is irreflexive and symmetric – corresponding to the

**Axioms**:

| | |
|---|---|
| All substitution instances of propositional tautologies | |
| $\bigwedge_{l=1}^{n}(\bigvee_{r\in R_l} V_l = r)$ | Char.Prop.1 |
| $\bigwedge_{l=1}^{n}\bigwedge_{r\in R_l}(V_l = r \to \bigwedge_{s\in R_l\setminus\{r\}} \neg V_l = s)$ | Char.Prop.2 |
| $X(\varphi\to\psi)\to(X\varphi\to X\psi)^1$ | $\mathrm{K}_X$ |
| $@_i(\varphi\to\psi)\to(@_i\varphi\to @_i\psi)$ | $\mathrm{K}_@$ |
| $@_i\varphi\leftrightarrow\neg @_i\neg\varphi$ | $\mathrm{Selfdual}_@$ |
| $@_i i$ | $\mathrm{Ref}_@$ |
| $@_i @_j\varphi\leftrightarrow @_j\varphi$ | Agree |
| $i\to(\varphi\leftrightarrow @_i\varphi)$ | Introduction |
| $\langle X\rangle @_i\varphi\to @_i\varphi^1$ | Back |
| $(@_i\langle X\rangle j\wedge @_j\varphi)\to @_i\langle X\rangle\varphi^1$ | Bridge |
| $\langle U\rangle i$ | GM |

**Rules**:

| | |
|---|---|
| From $\varphi$ and $\varphi\to\psi$, infer $\psi$ | Modus ponens |
| From $\varphi$, infer $X\varphi^1$ | Necessitation of $X$ |
| From $\varphi$, infer $@_i\varphi$ | Necessitation of @ |
| From $@_i\varphi$, where $i$ does not occur in $\varphi$, infer $\varphi$ | Name |
| From $(@_i\langle X\rangle j\wedge @_j\varphi)\to\psi$, where $i\neq j$ and $j$ does not occur in $\varphi$ or $\psi$, infer $@_i\langle X\rangle\varphi\to\psi^1$ | Paste |

[1] Here $X$ denotes either $F$ or $U$.

**Fig. 1.** The Hilbert-style proof system of LSN.

**Axioms**:

| | |
|---|---|
| All axioms for LSN of Fig. 1 | |
| $[\mathcal{D}]V_l = r\leftrightarrow(\bigvee_{\varphi\in\Phi,\ \mathsf{post}(\varphi)(l)=r}\varphi)\vee(\neg(\bigvee_{\varphi\in\Phi,\ l\in dom(\mathsf{post}(\varphi))}\varphi)\wedge V_l = r)$ | Red.Ax.Prop. |
| $[\mathcal{D}]i\leftrightarrow i$ | Red.Ax.Nom. |
| $[\mathcal{D}](\varphi\wedge\psi)\leftrightarrow[\mathcal{D}]\varphi\wedge[\mathcal{D}]\psi$ | Red.Ax.$\wedge$ |
| $[\mathcal{D}]\neg\varphi\leftrightarrow\neg[\mathcal{D}]\varphi$ | Red.Ax.$\neg$ |
| $[\mathcal{D}]@_i\varphi\leftrightarrow @_i[\mathcal{D}]\varphi$ | Red.Ax.@ |
| $[\mathcal{D}]F\varphi\leftrightarrow F[\mathcal{D}]\varphi$ | Red.Ax.$F$ |
| $[\mathcal{D}]U\varphi\leftrightarrow U[\mathcal{D}]\varphi$ | Red.Ax.$U$ |
| $[\mathcal{D}][\mathcal{D}']\varphi\leftrightarrow[(\mathcal{D};\mathcal{D}')]\varphi$ | Red.Ax.$DD$ |

**Rules**:

All the rules for LSN of Fig. 1

For all dynamic transformations $\mathcal{D},\mathcal{D}'\in\mathsf{DT}$.

**Fig. 2.** The Hilbert-style proof system of LDSN.

undirected networks very often studied in social networks analysis.[9] As mentioned in Section 2.1 irreflexivity and symmetry can be expressed (by pure formulas) in LSN. Thus, adding all substitution instances of these formulas allows us to derive a complete axiomatization of the logic of undirected networks.

### *3.2. Complete axiomatization of* LDSN

We now move on to give a complete axiomatization of the full dynamic logic LDSN. The axiomatization is shown in Fig. 2. The new axioms, referred to as *reduction axioms*, allow us to reduce all talk about dynamic properties of the network models to talk about their static properties. Moreover, they give us a better understanding of the dynamic transformations. For instance, the intuition behind the first reduction

[9] This is also the class of networks, representing "friendship", which is considered by [44,29].

axiom Red.Ax.Prop. is that if the variable $V_l$ is assigned the value $r$ after the dynamic transformation $\mathcal{D}$, then before the transformation either i) one of the post-conditions of $\mathcal{D}$ that specify a change resulting in $V_l = r$, is satisfied, or ii) no precondition of $\mathcal{D}$ that specify a change to the value taken by variable $V_l$ is satisfied and $V_l = r$ is already true.

The intuition behind the axiom Red.Ax.Nom. is that dynamic transformations do not change the names of agents. The axiom Red.Ax.$\wedge$ says that dynamic transformations commute with conjunction, while the axiom Red.Ax.$\neg$ says that they also commute with negation. That negation commutes with a dynamic modality might seem a bit surprising to readers familiar with public announcement logic or traditional dynamic epistemic logic. However, as dynamic transformations can always be executed (as discussed after Definition 9), $\neg\varphi$ is true after the dynamic transformation $\mathcal{D}$ if, and only if, it is not the case that $\varphi$ is true after a dynamic transformation $\mathcal{D}$. The axioms Red.Ax.@, Red.Ax.$F$, and Red.Ax.$U$ further state that the modalities $@_i$, $F$, and $U$ commute with dynamic transformations. The fact that dynamic transformation modalities commute with the other modalities highlights the fact that dynamic transformations of network models can be reduced to local changes at each agent in the network models. As such, the reduction axioms provide new insights about the behavior of dynamic transformations.

The way we will show completeness of this proof system is the usual way in Dynamic Epistemic Logic, namely by providing a truth-preserving translation from $\mathsf{LDSN}$ into $\mathsf{LSN}$. Before this, however, we need to define the composition of two dynamic transformations[10] as used in the last reduction axiom of Fig. 2.[11]

**Definition 13** *(Composition of dynamic transformations).* Given two dynamic transformations $\mathcal{D} = (\Phi, \mathsf{post})$ and $\mathcal{D}' = (\Phi', \mathsf{post}')$, the composition $(\mathcal{D}; \mathcal{D}') = (\Phi'', \mathsf{post}'')$ is such that

$$\Phi'' = \{\varphi \wedge [\mathcal{D}]\psi \mid \varphi \in \Phi, \psi \in \Phi'\} \cup \left\{\varphi \wedge [\mathcal{D}]\left(\bigwedge_{\psi\in\Phi'} \neg\psi\right) \,\middle|\, \varphi \in \Phi\right\}$$
$$\cup \left\{\left(\bigwedge_{\varphi\in\Phi} \neg\varphi\right) \wedge [\mathcal{D}]\psi \mid \psi \in \Phi'\right\},$$

and $\mathsf{post}''$ is such that

$$\begin{aligned}
\mathsf{post}''\big(\varphi \wedge [\mathcal{D}]\psi\big)(l) &= \mathsf{post}'(\psi)(l), \quad \text{if } l \in dom\big(\mathsf{post}'(\psi)\big)\\
\mathsf{post}''\big(\varphi \wedge [\mathcal{D}]\psi\big)(l) &= \mathsf{post}(\varphi)(l), \quad \text{if } l \in dom\big(\mathsf{post}(\varphi)\big) \setminus dom\big(\mathsf{post}'(\psi)\big)\\
\mathsf{post}''\left(\varphi \wedge [\mathcal{D}]\left(\bigwedge_{\psi\in\Phi'} \neg\psi\right)\right)(l) &= \mathsf{post}(\varphi)(l), \quad \text{if } l \in dom\big(\mathsf{post}(\varphi)\big)\\
\mathsf{post}''\left(\left(\bigwedge_{\varphi\in\Phi} \neg\varphi\right) \wedge [\mathcal{D}]\psi\right)(l) &= \mathsf{post}'(\psi)(l), \quad \text{if } l \in dom\big(\mathsf{post}'(\psi)\big).
\end{aligned}$$

Note that this definition is well-defined as $\Phi''$ will consist of pairwise unsatisfiable formulas. Moreover, while this definition might seem a bit complicated, this is only due to the fact that we have to take into account the following three cases for a given agent:

[10] Note that we treat ";" purely as a semantic operation on dynamic transformations. One could also have included ";" directly in the syntax in Definition 7. However, we have chosen not to do so, as we only use this composition of dynamic transformations for proving completeness.

[11] At first sight the last reduction axiom Red.Ax.$DD$ might seem superfluous. However, as we define our translation from $\mathsf{LDSN}$ to $\mathsf{LSN}$ "outside–in" on formulas, we need this reduction axiom for composition. If one defines the translation "inside–out" on formulas one would instead need "replacement of equivalents". However, "replacement of equivalents" cannot be derived from the other axioms in standard axiomatizations of public announcement logic and we suspect it cannot be here either. For an excellent discussion of these subtle issues concerning axiomatizations of dynamic epistemic logics see [47].

(i) One of the formulas in $\Phi$ is satisfied at the agent and afterwards the agent satisfies one of the formulas in $\Phi'$
(ii) One of the formulas in $\Phi$ is satisfied at the agent, but after the dynamic transformation $\mathcal{D}$ the agent does not satisfy any formula in $\Phi'$
(iii) None of the formulas in $\Phi$ is satisfied at an agent, but after the dynamic transformation $\mathcal{D}$ the agent does satisfy one of the formula in $\Phi'$

These three cases give rise to the three sets in the definition of $\Phi''$. Moreover, these three cases give rise to different definitions of $\mathsf{post}''$. In the case of $(i)$, there are three additional sub-cases according to whether a) the partial assignment $\mathsf{post}'(\psi)$ specifies a change of a feature, or b) $\mathsf{post}'(\psi)$ does not specify a change of a feature, but $\mathsf{post}(\varphi)$ does, or c) neither a) nor b) is the case. The cases a) and b) are directly taken care of in the definition of $\mathsf{post}''$, whereas c) is indirectly taken care of by the fact that $\mathsf{post}''$ might be partial assignment.

The following lemma, about composition of dynamic transformations will be useful:

**Lemma 14.** *For every network model $\mathcal{M}$ and any two dynamic transformations $\mathcal{D}$ and $\mathcal{D}'$ we have that:*

$$\left(\mathcal{M}^{\mathcal{D}}\right)^{\mathcal{D}'} = \mathcal{M}^{(\mathcal{D};\mathcal{D}')} \tag{3}$$

Before proving completeness we need to check the soundness of the reduction axioms (which, of course, also gives us soundness of the proof system). This is ensured by the following lemma:

**Lemma 15.** *For all models $\mathcal{M} = (A, \asymp, g, \nu)$ and all $a \in A$, the following hold:*

$$\mathcal{M}, a \models [\mathcal{D}]V_l = r \quad \textit{iff} \quad \mathcal{M}, a \models \Big(\bigvee_{\varphi\in\Phi,\ \mathsf{post}(\varphi)(l)=r} \varphi\Big) \vee \Big(\neg\Big(\bigvee_{\varphi\in\Phi,\ l\in dom(\mathsf{post}(\varphi))} \varphi\Big) \wedge V_l = r\Big) \tag{4}$$

$$\mathcal{M}, a \models [\mathcal{D}]i \quad \textit{iff} \quad \mathcal{M}, a \models i \tag{5}$$

$$\mathcal{M}, a \models [\mathcal{D}]\neg\varphi \quad \textit{iff} \quad \mathcal{M}, a \models \neg[\mathcal{D}]\varphi \tag{6}$$

$$\mathcal{M}, a \models [\mathcal{D}](\varphi \wedge \psi) \quad \textit{iff} \quad \mathcal{M}, a \models [\mathcal{D}]\varphi \wedge [\mathcal{D}]\psi \tag{7}$$

$$\mathcal{M}, a \models [\mathcal{D}]@_i\varphi \quad \textit{iff} \quad \mathcal{M}, a \models @_i[\mathcal{D}]\varphi \tag{8}$$

$$\mathcal{M}, a \models [\mathcal{D}]F\varphi \quad \textit{iff} \quad \mathcal{M}, a \models F[\mathcal{D}]\varphi \tag{9}$$

$$\mathcal{M}, a \models [\mathcal{D}]U\varphi \quad \textit{iff} \quad \mathcal{M}, a \models U[\mathcal{D}]\varphi \tag{10}$$

$$\mathcal{M}, a \models [\mathcal{D}]\big[\mathcal{D}'\big]\varphi \quad \textit{iff} \quad \mathcal{M}, a \models @\big[\big(\mathcal{D};\mathcal{D}'\big)\big]\varphi \tag{11}$$

**Proof.** We only provide the proof of (4) and leave out the other cases. Let $\mathcal{M}^{\mathcal{D}}$ be $(A, \asymp, g, \nu')$, where $\nu'$ is defined as in (1). Then we have the following equivalences:

$$\begin{aligned} \mathcal{M}, a \models [\mathcal{D}]V_l = r \quad &\text{iff} \quad \mathcal{M}^{\mathcal{D}}, a \models V_l = r \\ &\text{iff} \quad \nu'(a)(l) = r. \end{aligned}$$

Note that, $\nu'(a)(l) = r$ is the case if, and only if, either there is a $\varphi \in \Phi$ such that $\mathcal{M}, a \models \varphi$ and $\mathsf{post}(\varphi)(l) = r$, or there is no such $\varphi$, but $\nu(a)(l) = r$. Now, the first disjunct of this disjunction is equivalent to $\mathcal{M}, a \models (\bigvee_{\varphi\in\Phi,\mathsf{post}(\varphi)(l)=r} \varphi)$ while the second is equivalent to $\mathcal{M}, a \models (\neg(\bigvee_{\varphi\in\Phi,\ l\in dom(\mathsf{post}(\varphi))} \varphi) \wedge V_l = r)$. Hence,

$$
\begin{array}{lcl}
t(V_l = r) & = & V_l = r \\
t([\mathcal{D}]V_l = r) & = & t((\bigvee_{\varphi\in\Phi,\mathsf{post}(\varphi)(l)=r}\varphi) \vee (\neg(\bigvee_{\varphi\in\Phi,\ l\in dom(\mathsf{post}(\varphi))}\varphi) \wedge V_l = r))
\end{array}
$$

$$
\begin{array}{lcllcl}
t(i) & = & i & t([\mathcal{D}]i) & = & t(i) \\
t(\neg\varphi) & = & \neg t(\varphi) & t([\mathcal{D}]\neg\varphi) & = & t(\neg[\mathcal{D}]\varphi) \\
t(\varphi\wedge\psi) & = & t(\varphi)\wedge t(\psi) & t([\mathcal{D}](\varphi\wedge\psi)) & = & t([\mathcal{D}]\varphi \wedge [\mathcal{D}]\psi) \\
t(\Box\varphi) & = & \Box t(\varphi)^1 & t([\mathcal{D}]\Box\varphi) & = & t(\Box[\mathcal{D}]\varphi)^1 \\
 & & & t([\mathcal{D}][\mathcal{D}']\varphi) & = & t([(\mathcal{D};\mathcal{D}')]\varphi)
\end{array}
$$

[1] Here $\Box$ is either $F$, $@_i$, or $U$.

**Fig. 3.** The translation $t : \mathcal{L}_{\mathcal{DSN}} \to \mathcal{L}_{\mathcal{SN}}$.

$$\nu'(a)(l) = r \quad \text{iff} \quad \mathcal{M}, a \models \Big(\bigvee_{\varphi\in\Phi,\mathsf{post}(\varphi)(l)=r}\varphi\Big) \vee \Big(\neg\Big(\bigvee_{\varphi\in\Phi,\ l\in dom(\mathsf{post}(\varphi))}\varphi\Big) \wedge V_l = r\Big),$$

and (4) has been proven. $\Box$

The soundness of the axiomatization of LDSN now follows from the soundness of the axiomatization of LSN together with Lemma 15. To show completeness we first define a translation $t$ from $\mathcal{L}_{\mathcal{DSN}}$ into $\mathcal{L}_{\mathcal{SN}}$ as shown in Fig. 3. Note that the translation $t$ is not defined inductively on the usual notion of complexity of a formula. Therefore we cannot prove results regarding $t$ by induction on this complexity. However, the complexity of the formula immediately succeeding a dynamic transformation decreases through the translation, and we can use this fact. A new complexity measure $c$, such that $c$ decreases for every step of the translation, can be defined as follows:

**Definition 16** *(New complexity measure c).* Let the new complexity measure $c : \mathcal{L}_{\mathcal{DSN}} \cup \mathsf{DT} \to \mathbb{N}$, be defined as follows:

$$
\begin{aligned}
c(V_l = r) &= 1 \\
c(i) &= 1 \\
c(\neg\varphi) &= 1 + c(\varphi) \\
c(\Box\varphi) &= 1 + c(\varphi) \\
c(\varphi\wedge\psi) &= 1 + max\big(c(\varphi), c(\psi)\big) \\
c\big([\mathcal{D}]\varphi\big) &= \big(3\cdot|\Phi| + 3 + c(\mathcal{D})\big)\cdot c(\varphi) \\
c(\mathcal{D}) &= max\big\{c(\psi) \mid \psi\in\Phi\big\}
\end{aligned}
$$

where $\Box$ is "$@_i$", "$F$", or "$U$", and $\mathcal{D} = (\Phi, \mathsf{post})$.

We can show the following useful result: the translation of a dynamic formula can be reduced to the translation of a less complex formula:

**Lemma 17.** *For all $i \in \mathsf{NOM}$, all $V_l = r \in \mathsf{FP}$, all $\varphi, \psi \in \mathcal{L}_{\mathcal{DSN}}$, and all $\mathcal{D}, \mathcal{D}' \in \mathsf{DT}$ the following are true:*

1. $c([\mathcal{D}]i) > c(i)$
2. $c([\mathcal{D}]V_l = r) > c((\bigvee_{\varphi\in\Phi,\mathsf{post}(\varphi)(l)=r}\varphi) \vee (\neg(\bigvee_{\varphi\in\Phi,\ l\in dom(\mathsf{post}(\varphi))}\varphi) \wedge V_l = r))$
3. $c([\mathcal{D}]\neg\varphi) > c(\neg[\mathcal{D}]\varphi)$
4. $c([\mathcal{D}]\Box\varphi) > c(\Box[\mathcal{D}]\varphi)$

5. $c([\mathcal{D}](\varphi \wedge \psi)) > c([\mathcal{D}]\varphi \wedge [\mathcal{D}]\psi)$
6. $c([\mathcal{D}][\mathcal{D}']\varphi) > c([\mathcal{D}; \mathcal{D}']\varphi)$

The proof of this lemma is quite cumbersome and involves tedious computation. Thus, for space reasons, we do not include it here. This lemma allows us to prove that every formula of the logic LDSN is provably equivalent to its translation in LSN:

**Lemma 18.** *For all $\mathcal{L}_{\mathcal{DSN}}$ formulas $\varphi$,*

$$\vdash_{\mathsf{D}} \varphi \leftrightarrow t(\varphi) \tag{12}$$

**Proof.** The proof goes by induction on the new $c$-complexity. For $c(\varphi) = 1$, $\varphi$ is either of the form $V_l = r$ or of the form $i$. In both cases $\varphi = t(\varphi)$ and (12) is trivially satisfied. Now suppose that (12) holds for all $\varphi$ with $c(\varphi) \leq n$. Then, we need to prove that (12) holds for all $\varphi$ with $c(\varphi) = n + 1$. Thus, assume that $\varphi$ is a formula such that $c(\varphi) = n + 1$. We need to distinguish 4 cases: i) $\varphi$ is of the form $\neg\psi$; ii) $\varphi$ is of the form $\Box\psi$, with $\Box$ as in Definition 16; iii) $\varphi$ is of the form $\psi_1 \wedge \psi_2$; and iv)$\varphi$ is of the form $[\mathcal{D}]\psi$. We leave out the straightforward proofs of cases i), ii) and iii). To prove iv), we need to check the following sub-cases, corresponding to the 6 points of Lemma 17:

1. $\varphi$ is of the form $[\mathcal{D}]i$. By Lemma 17.1 and induction hypothesis, $\vdash_{\mathsf{D}} i \leftrightarrow t(i)$. By Red.Ax.Nom and the fact that $t(i) = t([\mathcal{D}]i)$, $\vdash_{\mathsf{D}} [\mathcal{D}]i \leftrightarrow t([\mathcal{D}]i)$.
2. $\varphi$ is of the form $[\mathcal{D}]V_l = r$. For readability, let us denote $(\bigvee_{\varphi\in\Phi,\mathsf{post}(\varphi)(l)=r}\varphi) \vee (\neg(\bigvee_{\varphi\in\Phi,\ l\in dom(\mathsf{post}(\varphi))}\varphi) \wedge V_l = r)$ by $\chi$. By Lemma 17.2 and induction hypothesis $\vdash_{\mathsf{D}} \chi \leftrightarrow t(\chi)$. By Red.Ax.prop and propositional logic, $\vdash_{\mathsf{D}} [\mathcal{D}]V_l = r \leftrightarrow t(\chi)$. Since $t([\mathcal{D}]V_l = r) = t(\chi)$, we conclude that $\vdash_{\mathsf{D}} [\mathcal{D}]V_l = r \leftrightarrow t([\mathcal{D}]V_l = r)$.
3. $\varphi$ is of the form $[\mathcal{D}]\neg\psi$. By Lemma 17.3 and induction hypothesis $\vdash_{\mathsf{D}} \neg[\mathcal{D}]\psi \leftrightarrow t(\neg[\mathcal{D}]\psi)$. By Red.Ax$\neg$ and propositional logic, $\vdash_{\mathsf{D}} [\mathcal{D}]\neg\psi \leftrightarrow t(\neg[\mathcal{D}]\psi)$ and since $t([\mathcal{D}]\neg\psi) = t(\neg[\mathcal{D}]\psi)$, we can conclude that $\vdash_{\mathsf{D}} [\mathcal{D}]\neg\psi \leftrightarrow t([\mathcal{D}]\neg\psi)$.
4. $\varphi$ is of the form $[\mathcal{D}]\Box\psi$. Similar to the case 3. just using Lemma 17.4 and the reduction axiom Red.Ax.$\Box$ instead.
5. $\varphi$ is of the form $[\mathcal{D}](\psi_1 \wedge \psi_2)$. Similar to the case 3. just using Lemma 17.5 and the reduction axiom Red.Ax.$\wedge$ instead.
6. $\varphi$ is of the form $[\mathcal{D}][\mathcal{D}']\psi$. Similar to the case 3. just using Lemma 17.6 and the reduction axiom Red.Ax.$DD$ instead. $\Box$

From Lemma 18 and the soundness of the proof system, it follows directly that all formulas are also semantically equivalent to their translation:

**Lemma 19.** *For all $\mathcal{L}_{\mathcal{DSN}}$ formulas $\varphi$, all models $\mathcal{M} = (A, \asymp, g, \nu)$, and all $a \in A$,*

$$\mathcal{M}, a \models \varphi \quad \Longleftrightarrow \quad \mathcal{M}, a \models t(\varphi)$$

Note that, translating pure formulas from $\mathcal{L}_{\mathcal{DSN}}$ results in pure formulas in $\mathcal{L}_{\mathcal{SN}}$. The general completeness result now follows:

**Theorem 2** *(Completeness for* LDSN*). Let $\Sigma$ be a substitution-closed set of pure $\mathcal{L}_{\mathcal{DSN}}$-formulas. Every set of $\mathcal{L}_{\mathcal{DSN}}$-formulas that is $\mathsf{D} + \Sigma$-consistent is satisfiable in a model whose underlying frame validates all the formulas in $\Sigma$.*

**Proof.** Assume that $\Gamma$ is $\mathsf{D} + \Sigma$-consistent. For a set of $\mathcal{L}_{\mathcal{DSN}}$-formulas $X$, let $t(X) := \{t(\varphi) \mid \varphi \in X\}$. Then $t(\Gamma)$ is $\mathsf{S} + t(\Sigma)$-consistent, for assume otherwise: Then there are $\varphi_1, ..., \varphi_n \in \Gamma$ such that $\vdash_{\mathsf{S}+t(\Sigma)} t(\varphi_1 \wedge ... \wedge \varphi_n) \rightarrow \bot$. But then also $\vdash_{\mathsf{D}+\Sigma} t(\varphi_1 \wedge ... \wedge \varphi_n) \rightarrow \bot$ (using Lemma 18 on formulas in $\Sigma$) and by Lemma 18, $\vdash_{\mathsf{D}+\Sigma} \varphi_1 \wedge ... \wedge \varphi_n \rightarrow \bot$, which is a contradiction to $\Gamma$ being $\mathsf{D} + \Sigma$-consistent. Now by Theorem 1, $t(\Gamma)$ is satisfiable in a model $\mathcal{M}$ (which is also a model for $\mathcal{L}_{\mathcal{DSN}}$), and by Lemma 19 it follows that $\Gamma$ is also satisfiable in $\mathcal{M}$.

Finally, for all pure formulas $\varphi \in \Sigma$, $t(\varphi)$ is a pure formula. Thus by Theorem 1 the underlying frame of $\mathcal{M}$ validates all of the formulas $t(\varphi) \in t(\Sigma)$. But by Lemma 19 the underlying frame then also validates all $\varphi \in \Sigma$. $\square$

## 4. Terminating tableau systems

In this section we add to the meta-theory of our static and dynamic logics by designing terminating tableau systems for them. Moreover, the tableau systems provide decision procedures for the logics which in turn let us automatically verify general properties of social network dynamics expressible in our logic. We start by providing a tableau system for the static logic LSN in Section 4.1 before moving on to a tableau system for the dynamic logic LDSN in Section 4.2.

### *4.1. A tableau system for* LSN

Let us start by explaining how tableau proofs work in general. The approach here is highly inspired by the work of Bolander and Blackburn [13,14]. In fact, our tableau system for LSN is identical to theirs with the exception of rules to deal with our special feature propositions. The proofs of termination, soundness, and completeness are also similar to the proofs in [13] and thus, the details are left out.

By a "tableau" we mean a downward branching tree where each node is labeled by a formula. The top node will be referred to as the "root" and the final bottom nodes will be referred to as "leaves". A branch is a finite path from the root to a leaf. Tableaux are expanded using tableau rules. The rules apply to branches and specify how a given branch can be expanded.

We assume a new countable infinite set of *prefixes* Pref. We will denote elements of Pref by $\sigma, \tau, \rho...$ and so on. The formulas labeling nodes in the tableaux will be *prefixed formulas* of the form $\sigma\varphi$, for a $\sigma \in$ Pref and $\varphi$ a $\mathcal{L}_{\mathcal{SN}}$-formula, or *accessibility formulas* of the form $\sigma \asymp \tau$ for $\sigma, \tau \in$ Pref.[12]

The tableau rules for the tableau system for LSN are given in Fig. 4. The rules are to be read in the following way: If a formula above the horizontal line occurs on a branch, then the branch can be expanded with a node(s) labeled by the formula(s) below the line. If more than one formula occurs above the line, all these formulas have to occur on a branch before the rule can be applied. If several formulas occur below the horizontal line separated by vertical lines, this means that the branch is split into several new branches each expanded with a node labeled by the given formula. Ignoring the accessibility formulas and formulas of the form $\sigma i$, the formula above the horizontal line in a rule will be called the *premise* of the rule and the formula(s) below the horizontal line the *conclusion(s)* of the rule.

The rules $(\neg F)$, $(@)$, $(\neg @)$ and $(\neg U)$ are called *prefix generation rules*. The construction of a tableau is done in the usual way with the constraints that no prefix generation rule is applied twice to the same premise on the same branch and a formula is never added to a branch if it already occurs on it. If one of the rules $(close1)$ or $(close2)$ have been applied to a branch, no other rules can be applied to that branch. A branch is called closed if one of the rules $(close1)$ or $(close2)$ have been applied to it, otherwise the branch

[12] The symbol "$\asymp$" was also used to represent the network structure in network models, but here we reuse it for accessibility formulas. Since the accessibility formulas are intended to specify the network structure of the model constructed in the completeness proof this reuse seems natural. Moreover, there will be no confusion as to when we are talking about neighbor agents in a network model or about accessibility formulas.

$$\frac{\sigma\neg i}{\tau i}(\neg)^1 \qquad \frac{\sigma\neg\neg\varphi}{\sigma\varphi}(\neg\neg) \qquad \frac{\sigma\varphi\wedge\psi}{\begin{array}{c}\sigma\varphi\\ \sigma\psi\end{array}}(\wedge) \qquad \frac{\sigma\neg(\varphi\wedge\psi)}{\sigma\neg\varphi \mid \sigma\neg\psi}(\neg\wedge)$$

$$\frac{\sigma @_i\varphi}{\begin{array}{c}\tau i\\ \tau\varphi\end{array}}(@)^1 \qquad \frac{\sigma\neg @_i\varphi}{\begin{array}{c}\tau i\\ \tau\neg\varphi\end{array}}(\neg @)^1 \qquad \frac{\sigma\neg U\varphi}{\tau\neg\varphi}(\neg U)^1 \qquad \frac{\sigma U\varphi}{\tau\varphi}(U)^2$$

$$\frac{\sigma\neg F\varphi}{\begin{array}{c}\sigma\asymp\tau\\ \tau\neg\varphi\end{array}}(\neg F)^1 \qquad \frac{\sigma F\varphi \quad \sigma\asymp\tau}{\tau\varphi}(F) \qquad \frac{\sigma\varphi \quad \sigma i \quad \tau i}{\tau\varphi}(Id)$$

$$\frac{}{\sigma V_l = r_1 \,|\, \sigma V_l = r_2 \,|\, \ldots \,|\, \sigma V_l = r_{k_l}}(prop.cut)^3$$

$$\frac{\sigma\varphi \quad \sigma\neg\varphi}{\mathsf{X}}(close1) \qquad \frac{\sigma V_l = r \quad \sigma V_l = r'}{\mathsf{X}}(close2)^4$$

[1] The prefix $\tau$ is new to the branch. [2] The prefix $\tau$ already occurs on the branch.

[3] Where $V_l$ and $\sigma$ already occur on the branch, and $R_l = \{r_1, r_2, \ldots, r_{k_l}\}$.

[4] Where $r \neq r'$.

**Fig. 4.** Tableau rules for the logic LSN.

is called open. A tableau is called closed if all its branches are closed, otherwise it is called open. A tableau proof of a formula $\varphi$ is a closed tableau with $\sigma\neg\varphi$ as the root formula (for some prefix $\sigma$).

Tableaux can been seen as non-deterministic searches for models that satisfy the root formula. Each branch represents such a possible model. In this way, the intuition behind the prefixes is that they represent agents in the models. Thus, a prefixed formula specifies what is "required" to be true in the models we are looking for and the accessibility formulas specify which accessibility relations have to hold between the worlds in the models.

To make the tableau system terminate, we add a *loop-check mechanism* as the one used in [13]. We first need the following two definitions:

**Definition 20.** For a branch $\Theta$ and a prefix $\sigma$ occurring on $\Theta$, we define the set:

$$T^{\Theta}(\sigma) := \{\varphi \mid \sigma\varphi \text{ occurs on } \Theta\}.$$

**Definition 21** *(Urfather).* Given a branch $\Theta$ and a prefix $\sigma$ occurring on it, the *urfather* of $\sigma$, written $u_{\Theta}(\sigma)$, is the earliest introduced prefix $\tau$ on $\Theta$ such that $T^{\Theta}(\sigma) \subseteq T^{\Theta}(\tau)$. A prefix $\sigma$ is an urfather on $\Theta$ if there is a prefix $\tau$ on $\Theta$ such that $\sigma = u_{\Theta}(\tau)$.

Now, in addition to the already mentioned constraints for constructing tableaux, we add the following loop-check constraint:

***(Loop-check)*** *A prefix generation rule is only allowed to be applied to a formula $\sigma\varphi$ on a branch if $\sigma$ is a urfather on that branch.*

With the addition of this constraint we can prove the following theorem:

**Proposition 1.** *Any tableau constructed using the given tableau system for* LSN *is finite.*

The proof of this proposition makes use of a common *subformula property* of tableau systems, namely that for all prefixed formulas $\sigma\varphi$ occurring on a tableau, $\varphi$ has to be a subformula of the root formula of the tableau. This heavily relies on the particular formulation of the tableau rules and the notion of a subformula, of course. For propositional logic this is enough to ensure that tableaux always will be of a finite size bounded by the root formula (as there can only be finitely many subformulas of a formula). However, with the presence of prefix generation rules, there can potentially by infinitely many prefixed formulas as new prefixes can be introduced. Still there can only be finitely many prefixed formulas $\sigma\varphi$ for each prefix $\sigma$ due to the subformula property. Thus, an infinite tableau would mean infinitely many different prefixes on a branch, which can be shown to yield a contradiction with the loop-check constraint. For the details of this argument, see [13] – with a slight modification of the notion of a subformula, due to our atomic feature propositions, the proof of finiteness of tableaux from [13] can be directly adopted for the logic LSN.

Now, given a formula $\varphi$ we can start a tableau for $\sigma\neg\varphi$ and keep applying rules until no more rules are applicable. This process is ensured to stop after finitely many steps according to Proposition 1. The resulting finite tableau will either be closed, in which case completeness will ensure that $\varphi$ is valid, or the tableau will be open, in which case the model construction of the completeness proof (see below) provides us with a counter-model to the validity of $\varphi$. Thus, we have a decision procedure for the logic LSN, and it follows that:

**Theorem 3** *(Decidability of* LSN*). The logic* LSN *is decidable.*

We now turn to soundness and completeness of the tableau system. Soundness is proved in the standard way for tableau systems and is straightforward by noticing that all the rules of Fig. 4 preserve satisfiability (in any given model). Then, if $\varphi$ is not valid, then $\neg\varphi$ is satisfiable and any tableau with $\sigma\neg\varphi$ at the root will contain at least one satisfiable branch, which in turn cannot be closed. Thus, no closed tableau for $\sigma\neg\varphi$ can exist and $\varphi$ cannot be provable. Completeness is also proved in the standard way. That is, from an open *saturated* tableau (–a tableau where all applicable rules have been applied) we construct a model that will be a model of the root formula. Now, if $\varphi$ is not provable then all tableaux for $\sigma\neg\varphi$ are open, especially one that is saturated. Thus, from this saturated open tableau we can construct a model of $\neg\varphi$, which further implies that $\varphi$ cannot be valid.

Constructing a model from an open branch of an open saturated tableau is the key step in the completeness proof together with proving that it actually is a model of the formulas on the branch. In constructing this model one has to be a bit careful as the set of prefixes cannot be used as the set of agents (as is often done) due to the loop-check condition. Instead, the set of agents will be all the urfarthers of the branch. Since prefix generation rules are not blocked by the loop-check condition for such prefixes we get closure under all the rules for urfarthers. With this, a truth lemma can be proven that states that whenever $\sigma\varphi$ occurs on the open saturated branch $\Theta$, $\varphi$ is satisfied at $u_\Theta(\sigma)$ in the model constructed. Formally, we have (for the details see [13]):

**Theorem 4** *(Completeness for* LSN*). If $\varphi$ is valid in* LSN*, then there is a tableau proof of $\varphi$.*

### *4.2. A tableau system for* LDSN

We extend the tableau system of Section 4.1 to a tableau system for LDSN. Moreover, we show that this extended tableau system is terminating, sound, and complete, as well. The approach in this section is highly inspired by the work of Hansen [23]. We first start by providing the tableau rules, we then show that the tableau system is terminating and we finally move on to the issues of soundness and completeness.

Before introducing the tableau system, it should be remarked that due to the translation of LDSN into LSN, and the sound, complete, and terminating tableau system for LSN just given, we could derive a decision procedure for LDSN by simply first translating any $\mathcal{L}_{\mathcal{DSN}}$ formula we want to check for validity into $\mathcal{L}_{\mathcal{SN}}$ and then use the decision procedure for LSN just specified. However, as argued for dynamic epistemic logics in [23], when searching for an actual tableau proof it can be an advantage to have a direct tableau system for LDSN instead of always going through the translation into LSN.

All the tableau rules for LSN, from Figure 4, plus the following rules:

$$\frac{\sigma[\mathcal{D}]\varphi}{\sigma T([\mathcal{D}]\varphi)}(\mathcal{D}) \qquad \frac{\sigma\neg[\mathcal{D}]\varphi}{\sigma\neg T([\mathcal{D}]\varphi)}(\neg\mathcal{D})$$

**Fig. 5.** Tableau rules for the logic LDSN.

Now, to formulate the tableau rules we will need the following "one step translation" of $\mathcal{L}_{\mathcal{DSN}}$ formulas of the form $[\mathcal{D}]\varphi$:

**Definition 22.** Define $T : \{[\mathcal{D}]\varphi \mid \mathcal{D} \in \mathsf{DT}, \varphi \in \mathcal{L}_{\mathcal{DSN}}\} \to \mathcal{L}_{\mathcal{DSN}}$, by:

$$\begin{aligned}
T\big([\mathcal{D}]V_l = r\big) &= \Big(\bigvee_{\varphi\in\Phi,\ \mathsf{post}(\varphi)(l)=r} \varphi\Big) \vee \Big(\neg\Big(\bigvee_{\varphi\in\Phi,\ l\in dom(\mathsf{post}(\varphi))} \varphi\Big) \wedge V_l = r\Big)\\
T\big([\mathcal{D}]i\big) &= i\\
T\big([\mathcal{D}](\varphi\wedge\psi)\big) &= [\mathcal{D}]\varphi \wedge [\mathcal{D}]\psi\\
T\big([\mathcal{D}]\neg\varphi\big) &= \neg[\mathcal{D}]\varphi\\
T\big([\mathcal{D}]@_i\varphi\big) &= @_i[\mathcal{D}]\varphi\\
T\big([\mathcal{D}]F\varphi\big) &= F[\mathcal{D}]\varphi\\
T\big([\mathcal{D}]U\varphi\big) &= U[\mathcal{D}]\varphi\\
T\big([\mathcal{D}]\big[\mathcal{D}'\big]\varphi\big) &= \big[\mathcal{D};\mathcal{D}'\big]\varphi
\end{aligned}$$

The rules for the tableau system for LDSN are shown in Fig. 5. What we have done is essentially to translate the reduction axioms of Fig. 2 into tableau rules and added them to the rules for LSN. In this way, we are translating formulas "on-the-fly" and only when needed (inspired by [23]).

We impose the same constraints on constructing tableaux as described in Section 4.1, including the loop-check condition. The termination proof will follow the structure of the proof of termination for LSN, as sketched in the previous section. However, we first need a new notion of "subformula" to ensure that all formulas in a tableau are subformulas of the root formula. The complexity measure $c$, defined in Definition 16, plays an important role here.

**Definition 23** *(c-subformula).* A formula $\psi$ is said to be a *c-subformula* of a formula $\varphi$ if the following are satisfied:

- $c(\psi) \leq c(\varphi)$
- Every nominal occurring in $\psi$ also occurs in $\varphi$

With this notion of c-subformulas, we can prove a couple of useful lemmas:

**Lemma 24.** *Assume that the set of nominals* NOM *is finite and let $k \in \mathbb{N}$. Then, there exist only finitely many distinct formulas $\varphi$ with $c(\varphi) \leq k$.*

**Proof.** Assume that the set of nominals NOM is finite and let $k \in \mathbb{N}$. Clearly, only finitely many formulas $\varphi$ without any dynamic transformation modality and with $c(\varphi) \leq k$ can be constructed (since we only have finitely many feature propositions and nominals). Denote this finite set $\mathcal{A}_0$. Then, only finitely many dynamic transformations $\mathcal{D} = (\Phi, \mathsf{post})$ such that $\Phi \subseteq \mathcal{A}_0$ exist. Now, allowing these finitely many dynamic transformation modalities $[\mathcal{D}]$ to occur, only finitely many new formulas $\varphi$ with $c(\varphi) \leq k$ can be constructed.

Denote this new set of formulas by $\mathcal{A}_1$. These formulas can now be used to define a new finite set of dynamic transformations $\mathcal{D} = (\Phi, \mathsf{post})$ with $\Phi \subseteq \mathcal{A}_1$. In this way, the recursive definition of $\mathcal{A}_m$ could easily continue and the sets $\mathcal{A}_m$ will all remain finite. Note that due to the requirement $c(\varphi) \leq k$, at some point we will not be able to construct any new dynamic transformations satisfying the requirement and thus, $\bigcup_{m\in\mathbb{N}} \mathcal{A}_m$ will be a finite set containing all formulas $\varphi$ with $c(\varphi) \leq k$. □

**Lemma 25.** *For every $\mathcal{L}_{\mathcal{DSN}}$ formula $\varphi$, the set of all c-subformulas of $\varphi$ is finite.*

**Proof.** This follows straightforwardly from the definition of a c-subformula, Lemma 24, and the fact that only finitely many nominals occur in $\varphi$. □

**Lemma 26.** *For every tableau rule, except the rules (prop.cut), (close1) and (close2), the c-complexity of its conclusion(s) is less than (or equal to) the c-complexity of its premise.*

**Proof.** The proof goes by inspection of all the tableau rules in Figs. 4 and 5. For the rules $(\mathcal{D})$ and $(\neg\mathcal{D})$ the fact follows from Lemma 17. □

We can now prove the desired subformula property:

**Lemma 27** *(c-subformula property). Let $\mathcal{T}$ be a tableau with $\sigma_0\varphi_0$ as the root formula. Then, for every prefixed formula $\sigma\varphi$ on $\mathcal{T}$, $\varphi$ is a c-subformula of $\varphi_0$.*

**Proof.** The proof goes by induction on the number of rule applications in $\mathcal{T}$. Since no rule can introduce any new nominals, it is clear that all nominals occurring on $\mathcal{T}$ must occur in the root formula $\varphi_0$. Moreover, by Lemma 26, no rule application can increase the c-complexity from the premise to a conclusion except for the rules (*prop.cut*), (*close*1) and (*close*2). However, (*close*1) and (*close*2) have no conclusions and any conclusion $\tau\psi$ of the (*prop.cut*) satisfies $c(\psi) = 1$, which is less than or equal to $c(\varphi_0)$ since $\varphi_0$ is a formula. For these reasons, it follows that a conclusion of any rule application is a c-subformula of $\varphi_0$ provided that the premise is a c-subformula of $\varphi_0$. □

This, together with Lemma 25, directly implies the following useful lemma:

**Lemma 28.** *Let $\Theta$ be a branch of a tableau and let $\sigma$ be a prefix occurring on $\Theta$. Then the set $T^{\Theta}(\sigma)$ is finite.*

With this lemma we can now follow the reasoning of [13] to obtain:

**Definition 29.** Let $\Theta$ be a branch. If a prefix $\tau$ has been introduced to the branch using a prefix generation rule on a formula of the form $\sigma\varphi$ we say that $\tau$ is generated by $\sigma$ and write $\sigma \prec_\Theta \tau$.

**Lemma 30.** *Let $\Theta$ be a tableau branch (constructed using the tableau system for* LDSN*). Then $\Theta$ is infinite if and only if there exists an infinite chain of prefixes:*

$$\sigma_1 \prec_\Theta \sigma_2 \prec_\Theta \sigma_3 \prec_\Theta \dots \ .$$

Again, adopting the proof from [13], we obtain:

**Proposition 2.** *Any tableau constructed using the given tableau system for* LDSN *is finite.*

Now, just as in Section 4.1, we obtain decidability:

**Theorem 5** *(Decidability of* LDSN*). The logic* LDSN *is decidable.*

Let us move on to soundness and completeness of the tableau system for LDSN. Soundness follows from the soundness of the tableau system for LSN together with the fact that the new rules $(\mathcal{D})$ and $(\neg\mathcal{D})$ preserve satisfiability, which again follows from Lemma 15. For completeness, we adopt the approach used to show completeness of the tableau system for LSN, as very briefly sketched in Section 4.1. Given an open saturated branch $\Theta$ of a tableau for LDSN, we define the canonical model $\mathcal{M}^\Theta = \langle A^\Theta, \asymp^\Theta, g^\Theta, \nu^\Theta \rangle$ by:

$$A^\Theta = \{\sigma \mid \sigma \text{ is an urfather on } \Theta\};$$
$$\asymp^\Theta = \{(\sigma, u_\Theta(\tau)) \in A^\Theta \times A^\Theta \mid \sigma \asymp \tau \text{ occurs on } \Theta\};$$
$$g^\Theta(i) = \begin{cases} \sigma, & \text{if } \sigma i \text{ occurs on } \Theta, \\ \sigma_0, & \text{if } \sigma i \text{ does not occur on } \Theta; \end{cases}$$
$$\nu^\Theta(\sigma)(l) = r \quad \text{iff} \quad \sigma V_l = r \text{ occurs on } \Theta;$$

where $\sigma_0$ is just some fixed element of $A^\Theta$.

Note that this model is well-defined. Now, the key step in the completeness of the tableau system is the following truth lemma:

**Lemma 31** *(Truth lemma). Let $\Theta$ be an open saturated branch of the tableau system. For any urfather $\sigma$ on $\Theta$ and any formula $\varphi$; if $\sigma\varphi$ occurs on $\Theta$, it holds that $\mathcal{M}^\Theta, \sigma \models \varphi$.*

**Proof.** The proof goes by induction on the $c$-complexity of $\varphi$. The cases when $\varphi$ is of the form $V_l = r$ or $\neg V_l = r$ follow directly from the definition of $\mathcal{M}^\Theta$. The cases for nominals, $\wedge$, $\neg$, $F$, and $U$ are as in [13].

The only new and interesting cases are when $\varphi = [\mathcal{D}]\psi$ or $\varphi = \neg[\mathcal{D}]\psi$. Assume that $\sigma[\mathcal{D}]\psi$ occurs on $\Theta$. Then by closure under the $(\mathcal{D})$ rule, $\sigma T([\mathcal{D}]\psi)$ also occurs on $\Theta$. However, $c(T([\mathcal{D}]\psi)) < c([\mathcal{D}]\psi)$ by Lemma 17 and it follows by induction that $\mathcal{M}^\Theta, \sigma \models T([\mathcal{D}]\psi)$. But then, by Lemma 15 it further follows that $\mathcal{M}^\Theta, \sigma \models [\mathcal{D}]\psi$. The case $\varphi = \neg[\mathcal{D}]\psi$ is analogous. □

With the truth lemma in place, completeness can now be ensured for the tableau system for LDSN:

**Theorem 6** *(Completeness for* LDSN*). If $\varphi$ is valid in* LDSN*, then there is a tableau proof of $\varphi$.*

## 5. Examples of applications

This section provides a few examples of the kind of modeling and reasoning about changes of distribution of features within social networks which LDSN allows for.

### *5.1. Pluralistic ignorance*

Pluralistic ignorance [34] is a phenomenon from social psychology which has been defined in various ways (see in [28,22,31,34,10]). We will stick to the definition from [10]: *a collective discrepancy between the agents' private attitudes and their public behavior*, a situation where all the individuals of a group have the same private attitude towards some proposition (say a belief in it), but publicly "display" a conflicting attitude towards it (say a belief in its negation).

For instance, consider a group of students. After a difficult lecture which none of the students actually understood, it may happen that none of them asks any question even though the teacher explicitly requested them to do so in case they did not understand the material. Even though none of the students actually understood the lecture, each of them believes that everybody else did. In addition to this simple classroom situation, real-life examples of pluralistic ignorance in the social and psychological literature include drinking habits among college students, attitudes towards norms of racial segregation, and many more [30,39,19,35].

In [16], we have studied pluralistic ignorance from a dynamic perspective and we have discussed how the social network structure constrains the dynamics of its dissolution. Our starting point was to note that such a phenomenon could not be modeled without distinguishing *two* properties of agents, their private belief state, which we call "inner belief" and their publicly observable behavior, which we call "expressed belief". As such, the phenomenon could not be captured by the "one property" framework for modeling belief change under conformity pressure offered by [29]. At the time, modeling pluralistic ignorance was our main motivation for designing a framework allowing to model the change of *several* properties of agents, and hence for adopting the "multi-property" approach which we continued pursuing in this paper. Our motivation here is much more general than the particular phenomenon of pluralistic ignorance, since we now consider any set of features of agents changing under local influence. However, we briefly recall below how to model the case of pluralistic ignorance, as an example of application of our general framework to a well-known dynamic social phenomenon.

Let two variables $V_I$ and $V_E$ correspond to the properties of "inner belief" (private mental state) and "expressed belief" (observable behavior), respectively. Each variable takes values from the same set: $R_I = R_E = \{b, n, u\}$, where $b$ represents belief in something (considered as given), $n$ represents the belief in its negation and $u$ represents the lack of belief in it and in its negation ("undecidedness").

To model how a situation will evolve, we need to assume some notion of social influence, that is, some dynamic transformation encoding how agents will change their belief states depending on the ones of their neighbors. One possibility, inspired by the (one-property) influence operator assumed in [29], is to consider that an agent is "brave enough" to express her actual private belief (i.e., $V_E = V_I$) at the next moment only when she has some "supporting" friend, i.e., some friends expressing what she privately believes or when she has no "conflicting" friends, i.e., no friend expressing a belief in the negation of what she privately believes.

Moreover, for simplicity, and to reflect the intuition that influence affects, at least in good part, the behavioral/visible/displayed side of agents, we consider that only their behavior is affected by what they observe, not their private belief state. What is important for us is that their behavior (what they display) depends on asymmetrical information: on the one hand, on what they themselves privately believe and, on the other hand, on what their friends/neighbors publicly express.

Let us define the corresponding dynamic transformation $\mathcal{D}_E = (\Phi_E, \mathsf{post}_E)$:

$$\begin{aligned}\Phi_E = \big\{ &\big(V_I = b \wedge \big(\langle F\rangle V_E = b \vee [F]V_E = u\big)\big) \vee [F]V_E = b,\\ &\big(V_I = n \wedge \big(\langle F\rangle V_E = n \vee [F]V_E = u\big)\big) \vee [F]V_E = n,\\ &V_I = u \wedge \neg[F]V_E = b \wedge \neg[F]V_E = n\big\}\end{aligned}$$

$$\begin{aligned}&\mathsf{post}_E\big(\big(V_I = b \wedge \big(\langle F\rangle V_E = b \vee [F]V_E = u\big)\big) \vee [F]V_E = b\big)(V_E) = b\\ &\mathsf{post}_E\big(\big(V_I = n \wedge \big(\langle F\rangle V_E = n \vee [F]V_E = u\big)\big) \vee [F]V_E = n\big)(V_E) = n\\ &\mathsf{post}_E\big(V_I = u \wedge \neg[F]V_E = b \wedge \neg[F]V_E = n\big)(V_E) = u\end{aligned}$$

Consider now a situation of pluralistic ignorance, in the sense that everybody privately believes something but expresses a belief in its negation. A model $\mathcal{M}$ is in a state of pluralistic ignorance if it satisfies the following formula:

$$PI\varphi := U(V_I = b \wedge V_E = n)$$

Now apply the transformation $\mathcal{D}_E$. Having another look at the preconditions set $\Phi_E$, note that none of them is satisfied at any agent. Therefore, none of the agents will change her behavior. In other words, a model in a state of pluralistic ignorance is stable. This corresponds to the intuition that pluralistic ignorance is a robust phenomenon: in the classroom example, unless some student does not obey the notion of influence defined by $\mathcal{D}_E$, no student will ask any question, and this no matter how long the teacher waits.

Now assume a model $\mathcal{M}$ slightly different: a unique agent, let it be named $i$, is expressing his private belief. Then the following is now satisfied in $\mathcal{M}$ and we will say that the model is in a state of "unstable pluralistic ignorance":

$$UPI := @_i(V_I = b \wedge V_E = b) \wedge U\big(\neg i \to (V_I = b \wedge V_B = n)\big).$$

It is easy to see that this situation is not stable under the transformation $\mathcal{D}_E$. For instance, considering the case of agent $i$ itself: $\mathcal{M}, i \vDash [F]V_E = n$ and therefore $\mathcal{M}, i \vDash (V_I = n \wedge (\langle F\rangle V_E = n \vee [F]V_E = u)) \vee [F]V_E = n$. Since we know that $\mathsf{post}_E((V_I = n \wedge (\langle F\rangle V_E = n \vee [F]V_E = u)) \vee [F]V_E = n)(V_E) = n$, agent $i$ will change his expressed belief state to a state in conflict with his private belief state, as a result of conformity pressure from all agents around him. But what about $i$'s neighbors? Consider an arbitrary agent $j$ such that $\mathcal{M}, i \vDash \langle F\rangle j$. Now $\mathcal{M}, j \vDash V_I = b \wedge \langle F\rangle V_E = b$ and therefore $\mathcal{M}, j \vDash (V_I = b \wedge (\langle F\rangle V_E = b \vee [F]V_E = u)) \vee [F]V_E = b$. Since we know that $\mathsf{post}_E((V_I = b \wedge (\langle F\rangle V_E = b \vee [F]V_E = u)) \vee [F]V_E = b)(V_E) = b$, agent $j$ will now have an expressed belief state in agreement with his private state. And similarly for any neighbor of the initiator $i$. Hence, agent $i$ and his neighbors have switched their expressed belief states after one application of the transformation. After one more step, $i$'s friends' friends will express their actual inner state, and then $i$'s friends' friends' friends, and so on. But then, by repeating the transformation $n$ times, all agents at distance less or equal to $n$ from $i$ will have changed their state at least once. Will such a cascading effect always reach a stable model? We have shown in [16] that this will depend on the network structure itself: if the network graph contains an odd cycle path (that is, if the graph is not two-colorable), then a (connected, symmetric and irreflexive) model in a state of unstable pluralistic ignorance will always stabilize and it will stabilize in a state where everybody expresses their actual private belief state, so pluralistic ignorance will be entirely "dissolved" or reversed. This reflects the intuition that pluralistic ignorance, in addition to being robust, is also fragile: one agent expressing his actual private belief state might turn everybody else![13]

### 5.2. *Diffusion of microfinance*

The fact that social network structures affect the adoption of new technologies has been well-documented for some time already. A classical example is the diffusion of hybrid seed corn among Iowa farmers [42] (–additionally, see the references in [7]). Still, the recent study [7] provides new insight about how social structures affect the spread of microfinance loans in small Indian villages. The authors of this study collected detailed data on various types of social ties and structures in 43 rural villages in Southern India before a microfinance institution entered the villages. Then, based on information from the microfinance institution, they compared the data on the social networks to the actual diffusion of microfinance loans in the villages.

It is argued in [7] that the diffusion of who is informed about the loaning possibilities is different from the diffusion of who chooses to participate in the microfinance loaning program. In the diffusion of microfinance, the most interesting parameter is who chooses to participate in the microfinance program. However, as shown by [7], this could not be estimated for individuals based on the participation of their neighbors in the social network. Moreover, the people who did not choose to participate in the microfinance program still passed on information about the program and thus, the diffusion of who was informed about the program did depend on whether an individual's neighbor was already informed (and chose to pass on the information). Hence, the two diffusion processes of information spreading and endorsement can come apart and as such the typical "SI Model" described in our introduction is not sufficient to represent such dynamics.

[13] For more details about the dissolution of pluralistic ignorance, a proof of the claim of stabilization, and an upper bound on such stabilization, we refer the reader to [16].

The spread of microfinance loans is a good example of why we might need *two* feature variables, one representing whether an individual is informed and one representing whether she has chosen to participate in the loaning program. The model presented in [7] is a probabilistic model and as such we cannot completely capture it in our framework. Nevertheless, we can describe some interesting variations. First, let us use two variables $V_I$ and $V_P$, where $V_I$ will keep track of who is informed about the microfinance program, and $V_P$ will keep track of who has actually chosen to take part in the program. As value set we will assume that $R_I = R_P = \{y, n\}$ for "yes" and "no", with the obvious interpretation that an agent satisfies $V_I = y$ if she is informed about the program and that she satisfies $V_P = n$ if she is not participating in the program.

One could imagine that an agent becomes informed about the microfinance program as soon as one of her friends is either informed or has chosen to participate. However, [7] estimated that people participating in the program were much more likely to pass on information about it than non-participants. Still, the non-participants' passing on of the information could not be neglected either. Thus, an alternative principle could be that an agent becomes informed about the microfinance program if at least one of her friends is participating or all of her friends are already informed. This suggests the following dynamic transformation $\mathcal{D}_I = (\Phi_I, \mathsf{post}_I)$ of the diffusion of information about the program, where

$$\Phi_I = \big\{\langle F\rangle V_P = y \vee FV_I = y\big\}$$
$$\mathsf{post}_I\big(\langle F\rangle V_P = y \vee FV_I = y\big)(V_I) = y.$$

Concerning the diffusion of participation, [7] claims that, in their data at least, there is no endorsement effect and thus whether an agent chooses to participate in the microfinance program does not solely depend on whether her friends have chosen to participate. One could assume that participation depends on other properties of each agent, for instance whether she needs a loan, whether she has potential for using such a loan etc. Let us collect all such reasons into one feature variable $V_O$ representing whether an agents is open/responsive to a loan (assuming that $R_R = \{y, n\}$ as well). Another precondition for choosing to participate in the micro-loan program is of course that the agent is actually informed about it. Thus, the diffusion of participation might be modeled by a dynamic transformation $\mathcal{D}_P = (\Phi_P, \mathsf{post}_P)$, where

$$\Phi_P = \{V_I = y \wedge V_O = y\}$$
$$\mathsf{post}_P(V_I = y \wedge V_O = y)(V_P) = y.$$

With such dynamics we can, for instance, show that if agent $j$ is friend with $i$ and $i$ participates in the program, then after one step of the dynamics $\mathcal{D}_I$, $j$ will be informed, in other words:

$$\big(@_j\langle F\rangle i \wedge @_i V_P = y\big) \rightarrow [\mathcal{D}_I]@_j V_I = y.$$

A tableau proof of this formula is shown in Fig. 6.

Following this, one can show that after an additional step of the dynamics $\mathcal{D}_P$, $j$ will participate in the program as well. One can also prove more complex properties such as if there is a path of length three from $i$ to $j$ where all agents on the path (including $i$ and $j$), are open to participation in the program and if $i$ is initially informed, then after three steps of the dynamics, $j$ is participating in the program.

According to [7], participants in the microfinance program were seven times more likely to pass on information about the program than non-participants, while non-participants counted for a third of the passing on of information about the program. This suggests that individuals were informed by non-participants at a much higher rate than they would be if they needed all of their friends to be informed first. It might be natural to assume that an agent gets informed when more than a third of her friends are informed. This kind of preconditions based on thresholds are quite common in the models of network science. Our current logic cannot capture this. However, in the next section, we will briefly discuss extensions of our framework to capture such thresholds preconditions and other interesting traits.

| | | |
|---|---|---|
| 1. | $\sigma_0\neg((@_j\langle F\rangle i \wedge @_i V_P = y) \rightarrow [\mathcal{D}_I]@_j V_I = y)$ | |
| 2. | $\sigma_0 @_j\langle F\rangle i$ | $(\neg\neg)$ and $(\wedge)$ on 1. |
| 3. | $\sigma_0 @_i V_P = y$ | $(\neg\neg)$ and $(\wedge)$ on 1. |
| 4. | $\sigma_0\neg[\mathcal{D}_I]@_j V_I = y$ | $(\neg\neg)$ and $(\wedge)$ on 1. |
| 5. | $\sigma_0\neg @_j[\mathcal{D}_I]V_I = y$ | $(\neg\mathcal{D})$ on 4. |
| 6. | $\sigma_1 j$ | $(\neg @)$ on 5. |
| 7. | $\sigma_1\neg[\mathcal{D}_I]V_I = y$ | $(\neg @)$ on 5. |
| 8. | $\sigma_1\neg(((\langle F\rangle V_P = y \vee FV_I = y)$ $\vee(\neg(\langle F\rangle V_p = y \vee FV_I = y) \wedge V_I = y))$ | $(\neg\mathcal{D})$ on 7. |
| 9. | $\sigma_1\neg(\langle F\rangle V_P = y \vee FV_I = y)$ | $(\neg\neg)$ and $(\wedge)$ on 8. |
| 10. | $\sigma_1\neg(\neg(\langle F\rangle V_P = y \vee FV_I = y) \wedge V_I = y)$ | $(\neg\neg)$ and $(\wedge)$ on 8. |
| 11. | $\sigma_1\neg\langle F\rangle V_P = y$ | $(\neg\neg)$ and $(\wedge)$ on 10. |
| 12. | $\sigma_1\neg FV_I = y$ | $(\neg\neg)$ and $(\wedge)$ on 10. |
| 13. | $\sigma_2 j$ | $(@)$ on 2. |
| 14. | $\sigma_2\langle F\rangle i$ | $(@)$ on 2. |
| 15. | $\sigma_2 \asymp \sigma_3$ | $(\neg F)$ on 14. |
| 16. | $\sigma_3 i$ | $(\neg F)$ on 14. |
| 17. | $\sigma_4 i$ | $(@)$ on 3. |
| 18. | $\sigma_4 V_P = y$ | $(@)$ on 3. |
| 19. | $\sigma_3 V_P = y$ | $(Id)$ on 18., 17. and 16. |
| 20. | $\sigma_2\neg\langle F\rangle V_P = y$ | $(Id)$ on 11., 6. and 13. |
| 21. | $\sigma_2 F\neg V_P = y$ | $(\neg\neg)$ on 20. |
| 22. | $\sigma_3\neg V_P = y$ | $(F)$ on 21. and 15. |
| 23. | X | $(close1)$ on 19. and 22. |

**Fig. 6.** A tableau proof of $(@_j\langle F\rangle i \wedge @_i V_P = y) \rightarrow [\mathcal{D}_I]@_j V_I = y$.

## 6. Conclusion and further research

The main goal of the present paper was to offer a general framework to reason about diffusion phenomena in social networks, or in general, about locally determined dynamics of features distribution in social network structures. We have given a complete axiomatization and a terminating tableau system for this logic. Let us conclude by quickly describing the relation of this work to other work and possible extensions of the framework to contribute building the bridge between social network analysis and logic.

### 6.1. Related work

As mentioned in the introduction, our work is largely inspired by the two-dimensional hybrid setting designed by Seligman et al. in [44,49,29]. In line with this work, we use hybrid logic tools to describe the network structure explicitly within the logic and the "friendship" modality $F$ to quantify over network-neighbors. However, instead of a two-dimensional framework, we have proposed a less complex (but of course less expressive) logic, since we do not include the possible worlds structure underlying the knowledge or belief states of the agents. Nevertheless, we can model belief change exactly in the way discussed by Seligman et al. in [29] within our simpler one-dimensional logic, since in [29] belief is informally captured by the satisfaction of one out of three mutually exclusive atomic propositions: an agent either believes that $p$, or believes that $\neg p$, or neither believes that $p$ nor that $\neg p$. Thus, the account of belief change given by [29] corresponds to a "one property, three values, one (repeated) specific dynamic transformation" case of our framework. Our work generalizes this idea of value-change to many properties, many values, and many possible ways of locally defining the values-change. In this sense, our setting can be seen both as a generalization and as a simplification of the framework of [29].

Note that another simplification of the very same two-dimensional hybrid framework of the seminal [44] has been proposed by [41] already. However, while [41] keeps the multi-agent possible worlds epistemic structure and lets go of the social network dimension, considering that this dimension can be captured by atomic propositions, we make the very opposite design choice here: we let go of the epistemically possible states dimension and keep only the hybrid social network dimension, thus showing that the way belief or knowledge was treated by [44,29] can be formally encoded by some atomic propositions. By doing this, we gain in generality: LDSN can represent any such change of repartition of features over a structured group of agents, not exclusively a change of knowledge, belief, or preferences. Of course, this simplification has a cost: we lose the possibility of fully combining epistemic structures and network structures in a dynamic logic. However, such an account of a properly two-dimensional hybrid dynamic doxastic logic has also been proposed both in [43] and in [15]. While [43] models channel-based communication, [15] investigates how agents in a social network structure have to communicate in order to merge their beliefs, in the way defined by [5] for unstructured groups of agents.

While our work is motivated mainly by diffusion phenomena from social network sciences, some conceptually relevant work was also motivated by communication within structured groups of agents, using different tools: Dynamic Epistemic Logic events restricted to communication channels [40], information via interaction structures [36,1], communication channels in Interpreted Systems [38], and communication channels in a combination of Interpreted Systems and Dynamic Epistemic Logic [48].

### *6.2. A simple extension to numerical threshold modalities*

In this paper, we have illustrated how a simple modal logic can be used to reason about a large class of dynamic processes. While our logic has limitations, there are several possible extensions that can make it applicable to larger classes of models from social networks analysis. In this subsection, we briefly sketch how to extend our logic with “numerical threshold modalities”. The next subsection briefly discusses other possible extensions which we leave for future research.

In [44,29,49] the changes considered depend exclusively on whether “all” or “some” of an agent’s neighbors believe/prefer/know something. Thus, changes can be modeled based on the thresholds “at least one” or “all” of an agent’s friends satisfying something. Similarly, in what precedes, we have restricted our threshold dynamics to those which are definable using our language $\mathcal{L}_{\mathcal{DSN}}$. However, one can argue that many diffusion phenomena involve numerical thresholds. Consider an example where agents only adopt a new technology if at least 5 of their friends/network-neighbors have adopted it already. To capture this idea, we add *numerical threshold modalities*[14] $\langle \geq nF \rangle$ for any $n \in \mathbb{N}$, with the interpretation “at least $n$ of my network-neighbors...”. Formally, the semantics is given by:

$$\mathcal{M}, a \models \langle \geq nF \rangle \varphi \quad \text{iff} \quad \big|\{b \in A \mid b \asymp a \text{ and } \mathcal{M}, b \models \varphi\}\big| \geq n,$$

where $|B|$ denotes the cardinality of the set $B$.

Axiomatizing this extension of our logic turns out to be surprisingly straightforward due to our use of hybrid logic. We simply add the following modified versions of the Bridge axiom and the Paste rule shown in Fig. 7. With these, straightforward extensions of the Lindenbaum Lemma and the Truth Lemma, used in the completeness proof, yield completeness of our static logic LSN extended with the numerical threshold modalities. For our dynamic logic LDSN adding the numerical threshold modalities are now straightforward as well since the following reduction axiom is easily seen to be valid:

$$[\mathcal{D}]\langle \geq nF \rangle \varphi \leftrightarrow \langle \geq nF \rangle [\mathcal{D}]\varphi.$$

[14] What we call “numerical threshold modalities” here are known as “graded modalities” in the modal logic literature and date back to Kit Fine’s paper [20]. They are also known as “qualified number restrictions” in the Description Logic literature [3].

**Axiom**:
$(\bigwedge_{s=1}^{n}(@_i\langle F\rangle j_s \wedge @_{j_s}\varphi) \wedge \bigwedge_{1\leq s<t\leq n} @_{j_s}\neg j_t) \to @_i\langle\geq nF\rangle\varphi$ [1] $n$-Bridge

**Rule**:
From $(\bigwedge_{s=1}^{n}(@_i\langle F\rangle j_s \wedge @_{j_s}\varphi) \wedge \bigwedge_{1\leq s<t\leq n} @_{j_s}\neg j_t) \to \psi$, where $i \neq j_s$ and $j_s$ does not occur in $\varphi$ or $\psi$ (for all $s$), infer $@_i\langle\geq nF\rangle\varphi \to \psi$ [1] $n$-Paste

[1] Here $j_1, ..., j_n$ denote distinct nominals.

**Fig. 7.** The additional axioms and rules for the modalities $\langle\geq nF\rangle$.

We leave the case of a sound, complete, and terminating tableau system for future research, but postulate that in similar manners, it is easily obtainable from our existing tableau system for LDSN. Alternatively, a sound, complete, and terminating tableau system for graded hybrid logic with the global modality is given in [27].

### 6.3. Future research

In addition to the numerical threshold modalities just discussed, one can also consider "proportional threshold modalities". For instance, in the microfinance example of the previous section, it might be more natural to specify that an agent gets informed if one third of her friends are informed. Another example is the dynamics induced by coordination (or anti-coordination) games played in social networks where the threshold to consider will depend on the payoffs involved in the corresponding game [18, Ch. 19].

Formally, we could also add proportional modalities of the form $\langle\geq \frac{p}{q}F\rangle$ for $p, q \in \mathbb{N}$ with $p \leq q$, with the following semantics:

$$\mathcal{M}, a \models \left\langle\geq\frac{p}{q}F\right\rangle\varphi \quad \text{iff} \quad \frac{|\{b \in A \mid b \asymp a \text{ and } \mathcal{M}, b \models \varphi\}|}{|\{b \in A \mid b \asymp a\}|} \geq \frac{p}{q}.$$

In addition to the already mentioned example of microfinance, the extended logic could be used to reason about several standard network analysis issues. For instance, the relationships between the density of clusters of a network structure and the possibility of a complete diffusion or "cascades" under a given threshold (see e.g. Chapter 19 of [18] for a presentation of a theorem without the use of logic and [17] for the same result using logical tools considerably different from our logic). In contrast to the numerical threshold modalities previously discussed, adding the proportional threshold modalities $\langle\geq \frac{p}{q}F\rangle$ to our static logic LSN requires more work with respect to the axiomatization and tableau system. We simply cannot use the same trick with the Bridge axiom and the Paste rule. Moreover, the given semantics of $\langle\geq \frac{p}{q}F\rangle$ might not be the obvious choice in networks where some agents have infinitely many friends. However, for the case of $\frac{p}{q} = \frac{1}{2}$ a nice solution can be found in [37]. Note that, with respect to the dynamic extension (in the finite case) we still have a straightforward reductions axiom in form of the following validity:

$$[\mathcal{D}]\left\langle\geq\frac{p}{q}F\right\rangle\varphi \leftrightarrow \left\langle\geq\frac{p}{q}F\right\rangle[\mathcal{D}]\varphi.$$

Another possible extension of our logic, in line with [49], is to add the transitive closure operator $F^*$ of the modality $F$, with the following semantics:

$$\mathcal{M}, a \models F^*\varphi \quad \text{iff} \quad \text{for all } b \in A; a \asymp^* b \text{ implies } \mathcal{M}, b \models \varphi,$$

where $\asymp^*$ is the transitive closure of the relation $\asymp$.[15] This "community modality" quantifies over what [49] names an agent's "community", that is, the agent's friends, the agent's friends' friends, the agent's friends' friends' friends, etc. Such a modality, as mentioned earlier in Section 2, allow us to express that a network is strongly connected [8,9,21]. We have left out this modality in our current logic because it is not relevant to the main ideas we wish to convey. Moreover, it would complicate the axiomatization of Section 3 and the tableau system of Section 4 considerably.

Occasionally, what is of interest is the limit behavior of diffusion processes within social networks. To capture this, a second "transitive closure" modality that we could add to our framework is the transitive closure of the dynamic transformation $\langle\mathcal{D}\rangle$, with the following semantics:

$$\mathcal{M}, a \models \langle\mathcal{D}^*\rangle\varphi \quad \text{iff} \quad \text{there is a } k \in \mathbb{N}_0 \text{ such that } \mathcal{M}^{k\mathcal{D}}, a \models \varphi.$$

In Section 2, we discussed how to describe stability, but our language as such cannot capture stabilization. On the other hand, with the $\langle\mathcal{D}^*\rangle$ modality we can easily express that a network model $\mathcal{M}$ stabilizes under the dynamic transformation $\mathcal{D}$ by the following formula

$$\langle\mathcal{D}^*\rangle\varphi_{stable}(\mathcal{D}).$$

However, sometimes, limiting behavior can be reduced to other properties of the network structures. For instance, [44] gives a characterization of stable and stabilizing models for the particular transformation under consideration, while [16] reduces stabilization of some type of network models to the existence of an odd cycle in the underlying network structure.[16] Again, we have chosen not to include $\langle\mathcal{D}^*\rangle$ in our logic as it is not essential to the main ideas of this paper. Moreover, adding $\langle\mathcal{D}^*\rangle$ is again likely to complicate the axiomatization and the tableau system of our logic. On the one hand, a similar transitive closure operator of Public Announcement Logic leads to an undecidable logic [32]. On the other hand, it actually turns out that the transitive closure operator of the substitution modality in Dynamic Logic of Propositional Assignments can be completely eliminated [4]. Thus, the matter remains an open problem, which we leave for future research.

Finally, another common class of models in network science relies heavily on probabilistic or undeterministic change, i.e., features changing in a particular way with a given probability. Recall the "SI" and "SIR" models of epidemic behavior in social networks from the introduction. Such models may be more realistic when they consider the *probability* that an agent gets infected given the health states of his neighbors. Hence, a desire could be to enrich our logic with probabilistic tools, for instance dynamic probabilistic modalities. However, this will require substantial changes to our logic and we leave it as future work to be done by us or others.

**Acknowledgements**

We would like to thank Johan van Benthem, Fenrong Liu and Sonja Smets for suggestions and comments during the elaboration of this paper. We are also grateful to the anonymous reviewers of LORI-IV for their precious feedback on [16], on which the present paper partially builds. Moreover, we would like to thank the two reviewers of this journal version whose comments help shape the final paper. The research of Zoé Christoff leading to these results has received funding from the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013)/ERC Grant agreement no. 283963.

[15] Such transitive closure modalities are also fairly standard in Propositional Dynamic Logic (PDL) [25].

[16] Talking about limiting behavior of social network dynamics using transitive closure modalities is also done in [24] for a particular model of opinion dynamics in social networks. However, the framework of [24] differs considerably from the present framework as it is based on a Fuzzy Logic.

Jens Ulrik Hansen is sponsored by the Swedish Research Council (VR) through the project "Collective Competence in Deliberative Groups: On the Epistemological Foundation of Democracy".

## References


[1] K.R. Apt, A. Witzel, J.A. Zvesper, Common knowledge in interaction structures, in: Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge, TARK '09, New York, NY, USA, ACM, 2009, pp. 4–13.

[2] C. Areces, B. ten Cate, Hybrid logics, in: P. Blackburn, J. van Benthem, F. Wolter (Eds.), Handbook of Modal Logic, Elsevier, Amsterdam, 2007, pp. 821–868.

[3] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider, The Description Logic Handbook – Theory, Implementation and Applications, second edn., Cambridge University Press, UK, 2007.

[4] P. Balbiani, N. Herzig Troquard, Dynamic logic of propositional assignments: a well-behaved variant of PDL, in: Proceedings of the 2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science. LICS '13, IEEE Computer Society, Washington, DC, USA, 2013, pp. 143–152.

[5] A. Baltag, S. Smets, Protocols for belief merge: Reaching agreement via communication, Log. J. IGPL 21 (3) (2013) 468–487.

[6] A. Baltag, L.S. Moss, S. Solecki, The logic of public announcements, common knowledge and private suspicious, Technical report SEN-R9922, CWI, Amsterdam, 1999.

[7] A. Banerjee, A.G. Chandrasekhar, E. Duflo, M.O. Jackson, The diffusion of microfinance, Science 341 (6144) (2013).

[8] M.R.F. Benevides, L.M. Schechter, Modal expressiveness of graph properties, Electron. Notes Theor. Comput. Sci. 205 (April 2008) 31–47.

[9] M.R.F. Benevides, L.M. Schechter, Using modal logics to express and check global graph properties, Log. J. IGPL 17 (5) (2009) 559–587.

[10] J.C. Bjerring, J.U. Hansen, N.J.L.L. Pedersen, On the rationality of pluralistic ignorance, Synthese 191 (2014) 2445–2470.

[11] P. Blackburn, B. ten Cate, Pure extensions, proof rules, and hybrid axiomatics, Stud. Log. 84 (2) (2006) 277–322.

[12] P. Blackburn, M. de Rijke, Y. Venema, Modal Logic, Cambridge Tracts in Theoretical Computer Science, vol. 53, Cambridge University Press, United Kingdom, 2001.

[13] T. Bolander, P. Blackburn, Termination for hybrid tableaus, J. Log. Comput. 17 (3) (2007) 517–554.

[14] T. Bolander, P. Blackburn, Terminating tableau calculi for hybrid logics extending K, in: Proceedings of the 5th Workshop on Methods for Modalities, M4M-5 2007, in: Electron. Notes Theor. Comput. Sci., vol. 231, 2009, pp. 21–39.

[15] Z. Christoff, A logic for social influence through communication, in: E. Lorini (Ed.), Proceedings of Eleventh European Workshop on Multi-Agent Systems, EUMAS 2013, in: CEUR, vol. 1113, 2013, pp. 31–39.

[16] Z. Christoff, J.U. Hansen, A two-tiered formalization of social influence, in: D. Grossi, O. Roy, H. Huang (Eds.), Logic, Rationality, and Interaction, in: Lecture Notes in Computer Science, vol. 8196, Springer, Berlin/Heidelberg, 2013, pp. 68–81.

[17] Z. Christoff, R. Rendsvig, Dynamic logics for threshold models and their epistemic extension, in: Epistemic Logic for Individual, Social, and Interactive Epistemology (ELISIEM) Workshop, 2014 (extended version in preparation).

[18] D. Easley, J. Kleinberg, Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, New York, USA, 2010.

[19] J.M. Fields, H. Schuman, Public beliefs about the beliefs of the public, Public Opin. Q. 40 (4) (1976) 427–448.

[20] K. Fine, In so many possible worlds, Notre Dame J. Form. Log. 13 (4) (October 1972) 516–520.

[21] J. Gate, I.A. Stewart, The expressibility of fragments of hybrid graph logic on finite digraphs, J. Appl. Log. 11 (3) (2013) 272–288.

[22] J.R.B. Halbesleben, M.R. Buckley, Pluralistic ignorance: historical development and organizational applications, Manag. Decis. 42 (1) (2004) 126–138.

[23] J.U. Hansen, Terminating tableaux for dynamic epistemic logics, in: Proceedings of the 6th Workshop on Methods for Modalities, M4M-6 2009, in: Electron. Notes Theor. Comput. Sci., vol. 262, 2010, pp. 141–156.

[24] J.U. Hansen, Reasoning about opinion dynamics in social networks, in: Proceedings of the Eleventh Conference on Logic and the Foundations of Game and Decision Theory, LOFT XI, Bergen, Norway, July 2014, pp. 27–30.

[25] D. Harel, D. Kozen, J. Tiuryn, Dynamic Logic, The MIT Press, Cambridge, 2000.

[26] M.O. Jackson, Social and Economic Networks, Princeton University Press, 2010.

[27] M. Kaminski, S. Schneider, G. Smolka, Terminating tableaux for graded hybrid logic with global modalities and role hierarchies, in: M. Giese, A. Waaler (Eds.), Automated Reasoning with Analytic Tableaux and Related Methods, in: Lecture Notes in Computer Science, vol. 5607, Springer, Berlin/Heidelberg, 2009, pp. 235–249.

[28] D. Krech, R.S. Crutchfield, Theories and Problems of Social Psychology, McGraw-Hill, New York, 1948.

[29] F. Liu, J. Seligman, P. Girard, Logical dynamics of belief change in the community, Synthese 191 (11) (2014) 2403–2431.

[30] D.T. Miller, C. McFarland, Pluralistic ignorance; when similarity is interpreted as dissimilarity, J. Pers. Soc. Psychol. 53 (1987) 298–305.

[31] D.T. Miller, C. McFarland, When social comparison goes awry: the case of pluralistic ignorance, in: J. Suls, T. Wills (Eds.), Social Comparison: Contemporary Theory and Research, Erlbaum, Hillsdale, NJ, 1991, pp. 287–313.

[32] J.S. Miller, L.S. Moss, The undecidability of iterated modal relativization, Stud. Log. 79 (3) (2005) 373–407.

[33] M.E.J. Newman, Networks: An Introduction, Oxford University Press, 2010.

[34] H.J. O'Gorman, The discovery of pluralistic ignorance: an ironic lesson, J. Hist. Behav. Sci. 22 (October 1986) 333–347.

[35] H.J. O'Gorman, S.L. Garry, Pluralistic ignorance – a replication and extension, Public Opin. Q. 40 (4) (1976) 449–458.

[36] E. Pacuit, R. Parikh, Reasoning about communication graphs, in: J. van Benthem, D. Gabbay, B. Löwe (Eds.), Interactive Logic: Selected Papers from the 7th Augustus de Morgan Workshop. Texts in Logic and Games, Amsterdam University Press, 2008, pp. 135–157.

[37] E. Pacuit, S. Salame, Majority logic, in: D. Dubois, C. Welty, M.A. Williams (Eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference, KR2004, AAAI Press, 2004, pp. 598–605.
[38] R. Parikh, R. Ramanujam, A knowledge based semantics of messages, J. Log. Lang. Inf. 12 (4) (2003) 453–467.
[39] D.A. Prentice, D.T. Miller, Pluralistic ignorance and alcohol use on campus: some consequences of misperceiving the social norm, J. Pers. Soc. Psychol. 64 (2) (1993) 243–256.
[40] F. Roelofsen, Exploring logical perspectives on distributed information and its dynamics, Master's thesis, University of Amsterdam, 2005.
[41] J. Ruan, M. Thielscher, A logic for knowledge flow in social networks, in: D. Wang, M. Reynolds (Eds.), AI 2011: Advances in Artificial Intelligence, in: Lecture Notes in Computer Science, vol. 7106, Springer, Berlin/Heidelberg, 2011, pp. 511–520.
[42] B. Ryan, N.C. Gross, The diffusion of hybrid seed corn in two Iowa communities, Rural Sociol. 8 (1) (1943) 15–24.
[43] K. Sano, S. Tojo, Dynamic epistemic logic for channel-based agent communication, in: K. Lodaya (Ed.), Logic and Its Applications, in: Lecture Notes in Computer Science, vol. 7750, Springer, Berlin/Heidelberg, 2013, pp. 109–120.
[44] J. Seligman, F. Liu, P. Girard, Logic in the community, in: M. Banerjee, A. Seth (Eds.), Logic and Its Applications, in: Lecture Notes in Computer Science, vol. 6521, Springer, Berlin/Heidelberg, 2011, pp. 178–188.
[45] J. van Benthem, Logical Dynamics of Information and Interaction, Cambridge University Press, The Netherlands, 2011.
[46] H. van Ditmarsch, W. van der Hoek, B. Kooi, Dynamic Epistemic Logic, Syntese Library, vol. 337, Springer, The Netherlands, 2008.
[47] Y. Wang, Q. Cao, On axiomatizations of public announcement logic, Synthese 190 (1) (2013) 103–134.
[48] Y. Wang, F. Sietsma, J. van Eijck, Logic of information flow on communication channels, in: D. Grossi, L. Kurzen, F.R. Velzques-Queseda (Eds.), Logic and Interactive Rationality, Yearbook 2009, ILLC, Amsterdam, 2010.
[49] Zhen L., J. Seligman, A logical model of the dynamics of peer pressure, Electron. Notes Theor. Comput. Sci. 278 (0) (2011) 275–288.