Dynamic Social Networks Logic

Zoé Christoff*

Institute for Logic, Language and Computation, University of Amsterdam

Jens Ulrik Hansen*

Department of Philosophy, Lund University

Abstract

We introduce a general framework for reasoning about dynamic processes within social networks, such as diffusion phenomena. First, we define the new Dynamic Social Networks Logic, a dynamic extension of standard hybrid modal logic. We then provide a complete axiomatization for this logic and give a terminating and complete tableau system for it. Finally, we show how to apply the framework to dynamic processes documented in social networks analysis.

Keywords: Modal Logic, Hybrid Logic, Social Networks Analysis, Diffusion in Social Networks

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 [1, 2, 3]). A variety of aspects of networks have been studied, such as: 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, behaviors, etc. What typically characterizes such processes is that their dynamics is local: whether an agent adopts a behavior/opinion/disease/product/trend depends on whether the agents related to him within the social network have adopted it already. Since social networks are graphs and since the dynamics depends on *local* properties, it seems like a natural choice to develop a dynamic *modal* logic to reason about such dynamics in social networks. This is exactly what this paper does.

To make clear the type of phenomena our framework is designed for, let us start by considering an example: the diffusion of a disease within a population.

*Corresponding author (*Zoe.Christoff@gmail.com*, *Jens_Ulrik.Hansen@fil.lu.se*)

Assume that each agent of the population is in either of two states: *infected* with the disease or *susceptible* to it. This type of models is commonly called "the SI Model" in the network literature [1]. Moreover, assume that the disease can only be contracted by being in contact with an infected agent. Consider the social 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 structured population. We first need to determine how contagious the disease is. So assume that each agent related to an infected agent in the network will get infected too at the next moment. This means that if some agent is infected to start with, then all agents related to him will get 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 would spread and what measures would be needed to contain it – the shortest network-path from an agent a to the initially infected agent i determines after how long agent a will get infected. In this example, the dynamics is essentially captured by the following local transformation rule: If any of your neighbors is infected, become infected yourself at the next moment of time.

The long term dynamics of such a contagion phenomena can also be investigated. Assuming that once an agent gets infected she will stay infected forever, each (connected finite component of the) network will reach a *stable* state where everybody is infected. However, one could very well imagine a different transformation rule: after being infected at one moment, an agent immediately recovers and becomes susceptible 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. The health status of an agent could take one of *three* values instead of two: *infected, susceptible, recovered* (the so-called "SIR Model", see for instance [1]). After being infected, agents can become "recovered", which might mean that they have become immune to the disease (alternatively, in some variants of the model, that they died from it) or that they will move back to being susceptible.

In the above, only the current health status of the agents matters. However, other features of agents may very well interfere with how their health status will change. For instance, imagine a genetic mutation such that agents carrying the mutation are immune to the disease or stay infected for longer. In this case, the epidemic dynamics reveals more complexity, and the transformation rule needs to combine several properties of agents.

Two aspects of the above examples will be particularly relevant to this paper. First of all, 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, has the genetic mutation, 34 years old, female, redhead, etc. 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 150, male or female, colors, etc. This assignment of values to properties will be captured by a particular kind of atomic propositions in our logic.

The second thing to remark is that the dynamics is defined in a purely local way. In the above, an agent change 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.

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[1, 2, 3]). In parallel, the dynamics of interacting agents has also been thoroughly addressed within the field of logic [4, 5, 6]. 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 [7, 8, 9, 10, 11, 12, 13, 14, 15]. 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 dynamics allowing to model complex locally governed dynamic phenomena within social network structures. Secondly, the paper provides a sound and complete axiomatization as well as a sound, complete and terminating tableau system for the new Dynamic Social Network Logic. In addition to providing deeper insights into the logic itself, we also discuss its applications to real examples from the network science.

The paper builds on some of our earlier work: the idea of equipping agents with different properties with different values was developed in [12], 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 [12] as a way of significantly generalizing the simple belief dynamics of [11].[2] However, contrary to the more philosophical [12], which focuses more on a particular case study, this paper contains a general and technical presentation of the logic, as well as an axiomatization and a terminating tableau system for it, together with an exemplification of its applications.

On the technical side, our logic will be an extension of standard hybrid logic [16] since hybrid logic provides relevant additional expressive powers compared

[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 [5, 6].

to basic modal logic, such as naming agents and defining more structural properties at the level of frames, as already noted by [7]. The axiomatization of our underlying static logic is very similar to the axiomatizations of [17, 16] and the additional axiomatization of our full dynamic logic borrows the reduction technique from [5, 6]. The tableau system for the static logic is adapted from [18], while the tableau system for our dynamic logic is inspired by the technique of [19].

The outline of the paper is as follows: Section 2 defines the Dynamic Social Networks Logic. The static part of the logic is introduced first and *dynamic transformation* are then added to it to obtain the full Dynamic Social Networks Logic. Section 3 provides a complete axiomatization of the logic, while section 4 provides a complete and terminating tableau system. 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 [18]. We have therefore included the proofs of termination and completeness for the *static* logic in Appendix B, in order to make the paper self-contained. 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 further research.

2. Dynamic Social Networks Logic

In this section, we introduce a hybrid logic to reason about the dynamics of properties of agents within social networks. We start with the static part of the logic, which we call *Static social networks logic*, to model the states of agents in a network at a given moment. We then move on to the full *Dynamic social networks logic* to represent the evolution of such situations.

2.1. Static social networks logic

As in [7] we will include standard tools from hybrid logic [16] to be able to talk about the network structure of agents. Hence, following [7], 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 them as having certain features that are instantiations of some fixed properties in consideration. Instead of having a set of standard propositional variables, we use equational statements to talk about features of agents in networks. We assume that each agent has n different relevant properties, to each of which is assigned one value from a finite set of possible values. To avoid any later confusion, let us first make precise the vocabulary we will be using: we will use the term "property" to refer to for instance age, gender, health status, etc., and we will use 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, throughout the rest of this section we will assume 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 $l \in \{1, \ldots, n\}$ *and some* $r \in R_l$*. The set of all feature propositions (for fixed sets of variables and values) will be denoted* FP*.*

The intuition is that the proposition $V_l = r$ is "true" (–the notion of "truth" will be defined in a moment) 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 that an agent is infected.[3]

In addition to the finite set of feature propositions (FP), we will assume a countable infinite set of nominals (NOM) used as names for agents in networks, just as nominals are used to refer to possible states in traditional hybrid logic [16]. We can now give the syntax for static social network language:

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

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

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 $\bigwedge_{i=1}^{n} \varphi_i := \bigwedge_{\varphi \in \{\varphi_1, \ldots, \varphi_n\}} \varphi := (\ldots((\varphi_1 \wedge \varphi_2) \wedge \varphi_3) \wedge \ldots) \wedge \varphi_n)$ and $\bigvee_{\varphi \in \{\varphi_1, \ldots, \varphi_2\}} := (\ldots((\varphi_1 \vee \varphi_2) \vee \varphi_3) \vee \ldots) \vee \varphi_n)$. The intuitive meaning of a formula $F\varphi$ is that "φ is true of all my friends" and the intuitive meaning of a formula $@_i\varphi$ is that "φ 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 "φ is true of all agents in the network".

As previously mentioned, we have assumed that a fixed set of feature propositions FP is given and 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 our language is relative to, we will use the notation $\mathcal{L}_{\mathcal{SSN}}(\mathsf{FP})$.

Before defining the semantics, let us first introduce the notion of assignment and define our models. Intuitively, our assignments will assign specific values to

[3] 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. $\neg P$).

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 from* $\{1, ..., n\}$ *to* $\prod_{l=1}^{n} R_l$*. 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 way to think about assignments 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* ν *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.

With the notion of a model we can now give semantics for the language $\mathcal{L}_{\mathcal{SSN}}$:

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

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

We say that a formula φ *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* φ *(taking* $\mathcal{M}$ *to be given). Two formulas* φ *and* ψ *are said to be* pairwise unsatisfiable *if* $\varphi \wedge \psi$ *is unsatisfiable. Given a model* $\mathcal{M} = (A, \asymp, g, \nu)$ *and a formula* φ *we write* $\mathcal{M} \models \varphi$ *if* $\mathcal{M}, a \models \varphi$ *for all* $a \in A$*. A formula* φ *is said to be* valid with respect to a class of frames

[4] In defining the social network models below, in which we will interpret our language, we actually only need the notion of full assignments. However, having partial assignments will simplify things when we define our dynamic social networks logic in sub section 2.2.

if $\mathcal{M} \models \varphi$ *for all models* $\mathcal{M}$ *build on some frame from the class. 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* SSNL *and referred to as the the static social networks logic.*

The static logic which we have defined so far allows us to talk about network *situations*, i.e., to describe the features of agents and the social network structure. In the next subsection, we will introduce the tools needed to talk about *changes* or *transformations* of such models, in order to represent the above-mentioned epidemic behaviors, for instance.

2.2. Dynamic transformations

We now extend our logic to deal with the dynamics of networks. We will make two important design choices. First, we are concerned exclusively with one particular type of changes within networks: the change of distribution of features 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 in this paper: agents are essentially just bundles of features with fixed names, and the question of *how* exactly such network models should change is so open-ended that we consider that the safest option is to offer a general framework which can allow for any such type of change, as long as it is locally definable in terms of our language. Therefore, 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 should be updated, i.e, how those features should be redistributed on the network. In other words, we are abstracting as much possible from particular diffusion examples given by the networks analysis literature by building a framework which can deal with them altogether.

What we want is a way to obtain a new model from a given model through some dynamic transformation. In this sense, the dynamic modalities which we will add to our language are comparable to the event models modalities of Dynamic Epistemic Logic [5, 6]. However, instead of the event models of Dynamic Epistemic Logic we will talk about *dynamic transformations*. On the syntactic level, we will add formulas on the form $[\mathcal{D}]\varphi$ for a given dynamic transformation $\mathcal{D}$.

Definition 6 (Dynamic transformations)**.** *A* dynamic transformation *is a pair* $\mathcal{D} = (\Phi, \mathsf{post})$ *consisting of a non-empty finite set* Φ *of pairwise unsatisfiable formulas (from the language* $\mathcal{L}_{\mathcal{DSN}}$ *to be defined in Definition 7)*[5] *and a post-*

[5] 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 Φ involving $\mathcal{D}$ itself. Nevertheless, we can allow formulas of $\mathcal{L}_{\mathcal{DSN}}$ in Φ constructed on an "earlier stage" in a simultaneous inductive definition of dynamics transformations and the language $\mathcal{L}_{\mathcal{DSN}}$. In

condition function $\mathsf{post} : \Phi \to \mathcal{V}$. *The set* Φ *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* φ.

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, φ 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 $dom(\mathsf{post}(\varphi))$.

In the following we assume a fixed set of dynamic transformations to be given and denote it by DT. We can now specify the syntax of our dynamic language:

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

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

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

As for the static language $\mathcal{L}_{\mathcal{SSN}}$, when we need to make it explicit which set of dynamic transformations DT (and feature propositions 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 semantics of formulas involving dynamic modalities transforms 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* ν' *is defined by:*

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

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 satisfies φ 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

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 [6, Ch. 6].

where there are no formulas in Φ 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 φ at a in $\mathcal{M}$ inductively as in Definition 5 with the additional clause:*

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

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 DSNL *and referred to as Dynamic social networks logic.*

Before moving on, let us consider the small example from the introduction concerning diffusion of a disease again. Here we might have to variables V_{HS} and V_{GM} keeping track of the health status of of the agents and whether they have the genetic mutation, i.e. $R_{HS} = \{susceptible, infected, recovered\}$ and $R_{GM} = \{yes, no\}$. Thus, $V_{HS} = susceptible \wedge V_{GM} = yes$ is true of an agent if she is susceptible to the disease and she has the genetic mutation. We could then specify the following dynamic transformation $\mathcal{D} = (\Phi, \mathsf{post})$, for instance:

Φ :	post :
$V_{HS} = susceptible \wedge V_{GM} = no \wedge \langle F \rangle V_{HS} = infected$	$\mathsf{post}(HS) = infected$
$V_{GM} = yes$	$\mathsf{post}(HS) = recovered$
$V_{HS} = infected$	$\mathsf{post}(HS) = recovered$

This dynamic transformation represents the fact that a susceptible agent without the genetic mutation becomes infected if at least one of her neighbors is infected, while an agent having the genetic mutation is immediately recovered and does not get infected. 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. More complex examples of applications of DSNL 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 agents from $\mathcal{A}$ are 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 $(\mathcal{M}, a)$. 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 φ. 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:*

$$\begin{aligned} \mathcal{M}^{0\mathcal{D}} &:= \mathcal{M} \\ \mathcal{M}^{(k+1)\mathcal{D}} &:= (\mathcal{M}^{k\mathcal{D}})^{\mathcal{D}}. \end{aligned}$$

Some interesting 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 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.[6] 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

[6] While our language can express stability, it cannot express stabilization. The straight forward way would be to add the PDL transitive closure construct $*$ to the modality $[\mathcal{D}]$. However, what would be interesting is to find a formula without the $\langle\mathcal{D}^*\rangle$ operator that defines that a network stabilizes, just as done in [8] for the case of a logic of preference change. We will leave this for future research, though. The transitive closure operator $*$ and stabilization is discussed in more details in the concluding Section 6.

stable under $\mathcal{D}$ by[7]:

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

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 structure. 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) language. Moreover, we have shown that our language can capture 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 change 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

Now that we have seen what can be modeled within our framework, let us see how we can use our logic to reason about the behavior of networks. In this section, we will provide sound and complete Hilbert-style proof systems for the logics of Section 2. The axiomatization we provide for the static logic SSNL follows that of [17, 16] with a few modifications, while the axiomatization we provide for the dynamic logic DSNL expands that of the static logic with "reduction axioms" – a standard technique of Dynamic Epistemic Logic [6].

We will start by giving the Hilbert-style axiomatization of SSNL, however, before this we need to fix some standard terminology for Hilbert-style proof systems: A proof of φ is a finite sequence of formulas ending with φ 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 SSNL and $\vdash_{\mathsf{D}}$ for provable in the proof-system for DSNL. In the following, X will thus stand for either S or

[7] Another way of expressing that a network model is stable would be to follow the line of [11]. 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.

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

[1] Here X denotes either F or U.

Figure 1: The Hilbert-style proof system of SSNL

D. For a set of formulas Γ, $\Gamma\vdash_{\mathsf{X}}\varphi$ 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 Σ, let $\mathsf{X}+\Sigma$ denote the logic obtained by adding all the formulas in Σ as axioms. That φ is provable in the logic $\mathsf{X}+\Sigma$ will then be denoted by $\vdash_{\mathsf{X}+\Sigma}\varphi$. A set of formulas Γ is said to be $\mathsf{X}+\Sigma$-inconsistent if $\Gamma\vdash_{\mathsf{X}+\Sigma}\bot$, and $\mathsf{X}+\Sigma$-consistent otherwise. A formula φ is pure if it does not contain any feature propositions. A set of formulas Σ is called substitution-closed if it is closed under uniform substitution of nominals by nominals.

3.1. Complete axiomatization of SSNL

The Hilbert-style axiomatization of SSNL is shown in Figure 1. As previously mentioned, the axiomatization in 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 at most one value.

Soundness and completeness of this type of axiomatization are also standard results in the hybrid logic literature (see [17, 16]). Thus, we will skip the proof of these properties and only state the completeness theorem:

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

We will briefly describe how this axiomatization can be extended to particular classes of network models. In particular, the completeness with pure formulas in Theorem 1 allows us to restrict our logic to certain classes of networks. For instance, we might want to restrict ourselves to networks where the relation $\asymp$ is symmetric, corresponding to undirected networks, or networks where $\asymp$ is irreflexive, i.e. no agent is connected to themselves. Now, it is a well-known fact [20] that symmetry can be defined by

$$i \to F\langle F\rangle i$$

and that irreflexivity can be defined by

$$i \to \neg\langle F\rangle i.$$

Thus, adding all substitution instances of the above formulas allows us to derive complete axiomatizations of the logic of symmetric network models as well as the logic of irreflexive network models (or the logic of both properties). Note that, the fact we are using a hybrid logic is essential here as irreflexivity cannot be defined in basic modal logic [20].

3.2. Complete axiomatization of DSNL

We now move on to give a complete axiomatization of the full dynamic logic DSNL. The axiomatization is shown in Figure 2. The new axioms are referred to as *reduction axioms*. In essence, these reduction axioms allow us to reduce all talk about dynamic properties to talk about static properties of the network models. Moreover, the reduction axioms give us a better understanding of the dynamics transformations. For instance, the intuition behind the first reduction axiom Red.Ax.Prop. is that if the variable V_l is assigned the value r after the dynamic transformation $\mathcal{D}$ then it can only be the case if 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 variable V_l is satisfied and $V_l = r$ is already true.

The intuition behind the reduction axiom Red.Ax.Nom. is simply that dynamic transformations do not change the names of agents. The reduction axiom Red.Ax.$\wedge$ says that dynamic transformations commutes with conjunction, while the reduction axiom Red.Ax.$\neg$ says that dynamic transformations also commutes 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), it is not the case that φ is true after a dynamic transformation $\mathcal{D}$ if, and only, if $\neg\varphi$ is true after the dynamic transformation $\mathcal{D}$. The reduction axioms Red.Ax.@, Red.Ax.F, and Red.Ax.U further state that the modalities $@_i$, F, and U also commute with dynamic transformations. The fact that dynamic transformation modalities commute with the other modalities highlight the fact that dynamic transformations of network models can be reduced to local changes at each agent in the

Axioms:

All axioms for SSNL of Figure 1

$[\mathcal{D}]V_l{=}r \leftrightarrow \big(\bigvee_{\varphi\in\Phi,\ \mathsf{post}(\varphi)(l)=r}\varphi\big) \vee \big(\neg(\bigvee_{\varphi\in\Phi,\ l\in dom(\mathsf{post}(\varphi))}\varphi) \wedge V_l{=}r\big)$	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 SSNL of Figure 1

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

Figure 2: The Hilbert-style proof system of DSNL

network models. As such, the reduction axioms provide new insights about the behavior of dynamic transformations. All the intuitions about the reduction axioms are further spelled out in the proof of Lemma 15.

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 DSNL into SSNL. Before this, however, we need to define the composition of two dynamic transformations as used in the last reduction axiom of Figure 2.[8]

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*

$$\begin{aligned}\Phi'' \quad = \quad & \{\varphi \wedge [\mathcal{D}]\psi \mid \varphi \in \Phi, \psi \in \Phi'\} \cup \{\varphi \wedge [\mathcal{D}]\big(\bigwedge_{\psi\in\Phi'} \neg\psi\big) \mid \varphi \in \Phi\} \\ & \cup \{\big(\bigwedge_{\varphi\in\Phi} \neg\varphi\big) \wedge [\mathcal{D}]\psi \mid \psi \in \Phi'\},\end{aligned}$$

[8] At first sight the last reduction axiom Red.Ax.DD might seem superfluous. However, as we define our translation from DSNL to SSNL "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". Though, "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 [21].

and post'' *is such that*

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

Note that this definition is well-defined as Φ'' 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:

(i) One of the formulas in Φ is satisfied at the agent and afterwards the agent satisfies one of the formulas in Φ'

(ii) One of the formulas in Φ is satisfied at the agent, but after the dynamic transformation $\mathcal{D}$ the agent does not satisfy any formula in Φ'

(iii) None of the formulas in Φ is satisfied at an agent, but after the dynamic transformation $\mathcal{D}$ the agent does satisfy one of the formula in Φ'

These three cases give rise to the three sets in the definition of Φ''. Moreover, these three cases give rise to different definitions of 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 post'', whereas c) is indirectly taken care of by the fact that post'' might be partial assignment.

For composition of dynamic transformations we have the following useful lemma, which is proven in Appendix A:

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

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

Now, to prove completeness we first need soundness of the reduction axioms (which, of course, also gives us soundness of the proof system). Soundness of the reduction axioms 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:*

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

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

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

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

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

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

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

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

Proof. We only provide the proof of (4). The rest of the cases can be found in Appendix A. Let $\mathcal{M}^{\mathcal{D}}$ be $(A, \asymp, g, \nu')$, where ν' is defined as in (1). Then we have the following equivalences:

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

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 φ, but $\nu(a)(l) = r$. Now, the first disjunct of this disjunction is equivalent to $\mathcal{M}, a \models \big(\bigvee_{\varphi\in\Phi,\mathsf{post}(\varphi)(l)=r}\varphi\big)$ while the second is equivalent to $\mathcal{M}, a \models \big(\neg(\bigvee_{\varphi\in\Phi,\ l\in dom(\mathsf{post}(\varphi))}\varphi) \wedge V_l = r\big)$. Hence,

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

and (4) has been proven. □

The soundness of the axiomatization of DSNL follows from the soundness of the axiomatization of SSNL together with Lemma 15. To show completeness we first define a translation t from $\mathcal{L}_{\mathcal{DSN}}$ into $\mathcal{L}_{\mathcal{SSN}}$. The definition of $t : \mathcal{L}_{\mathcal{DSN}} \to \mathcal{L}_{\mathcal{SSN}}$ is shown in Figure 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 trough the translation, and this we can use. A new complexity measure c can be defined such that c decreases for every step of the translation. The definition of the new complexity

$$
\begin{array}{lcl@{\qquad}lcl}
t(V_l = r) & = & V_l = r & & & \\
t([\mathcal{D}]V_l = r) & = & t\big(\big(\bigvee_{\varphi\in\Phi,\mathsf{post}(\varphi)(l)=r}\varphi\big) \vee \big(\neg(\bigvee_{\varphi\in\Phi,\ l\in dom(\mathsf{post}(\varphi))}\varphi) \wedge V_l = r\big)\big) & & & \\
t(i) & = & i & t([\mathcal{D}]i) & = & t(i) \\
t(\neg\varphi) & = & \neg t(\varphi) & t([\mathcal{D}]\neg\varphi) & = & t(\neg[\mathcal{D}]\varphi) \\
t(\varphi \wedge \psi) & = & t(\varphi) \wedge t(\psi) & t([\mathcal{D}](\varphi \wedge \psi)) & = & t([\mathcal{D}]\varphi \wedge [\mathcal{D}]\psi) \\
t(\Box\varphi) & = & \Box t(\varphi)^1 & t([\mathcal{D}]\Box\varphi) & = & t(\Box[\mathcal{D}]\varphi)\ ^1 \\
 & & & t([\mathcal{D}][\mathcal{D}']\varphi) & = & t([(\mathcal{D};\mathcal{D}')]\varphi)
\end{array}
$$

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

Figure 3: The translation $t : \mathcal{L}_{\mathcal{DSN}} \to \mathcal{L}_{\mathcal{SSN}}$.

measure looks 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:*

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

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

For this new complexity measure we can show the following useful result, namely that 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. $c([\mathcal{D}]V_l = r) > c\big(\big(\bigvee_{\varphi\in\Phi,\mathsf{post}(\varphi)(l)=r}\varphi\big) \vee \big(\neg(\bigvee_{\varphi\in\Phi,\ l\in dom(\mathsf{post}(\varphi))}\varphi) \wedge V_l = r\big)\big)$
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, we have included it in Appendix A. However, using this lemma and the new complexity measure allow us to prove that every formula of the logic DSNL is provably equivalent to its translation in SSNL:

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

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

Proof. The proof goes by induction on the new c-complexity. For $c(\varphi) = 1$, φ 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 φ with $c(\varphi) \leq n$. Then, we need to prove that (12) holds for all φ with $c(\varphi) = n+1$. Thus, assume that φ is a formula such that $c(\varphi) = n + 1$. We need to distinguish 4 cases:

i) φ is of the form $\neg\psi$. Then, by Def. 16, $c(\psi) = n$, and by induction hypothesis, $\vdash_{\mathsf{D}} \psi \leftrightarrow t(\psi)$. By propositional logic, $\vdash_{\mathsf{D}} \neg\psi \leftrightarrow \neg t(\psi)$ and given that $t(\neg\psi) = \neg t(\psi)$, it follows that $\vdash_{\mathsf{D}} \neg\psi \leftrightarrow t(\neg\psi)$.

ii) φ is of the form $\Box\psi$, with $\Box$ as in Def. 16, by which $c(\psi) = n$. By induction hypothesis, $\vdash_{\mathsf{D}} \psi \leftrightarrow t(\psi)$. By necessitation of $\Box$, $K_{\Box}$ rule and propositional logic, $\vdash_{\mathsf{D}} \Box\psi \leftrightarrow \Box t(\psi)$. Since $t(\Box\psi) = \Box t(\psi)$, it follows that $\vdash_{\mathsf{D}} \Box\psi \leftrightarrow t(\Box\psi)$.

iii) φ is of the form $\psi_1 \wedge \psi_2$. This is very similar to the case ii).

iv) φ is of the form $[\mathcal{D}]\psi$. We need to check the following sub-cases, corresponding to the 6 points of Lemma 17:

1. φ 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. φ 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 χ. 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. φ 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. φ 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. φ is of the form $[\mathcal{D}](\psi_1 \wedge \psi_2)$. Similar to the case 3. just using Lemma 17.5 and and the reduction axiom Red.Ax.$\wedge$ instead.
6. φ is of the form $[\mathcal{D}][\mathcal{D}']\psi$. Similar to the case 3. just using Lemma 17.6 and and the reduction axiom Red.Ax.DD instead.

□

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 φ, all models $\mathcal{M} = (A, \asymp, g, \nu)$, and all $a \in A$,*

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

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

Theorem 2 (Completeness for DSNL)**.** *Let Σ 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 Σ.*

Proof. Assume that Γ 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 $\vdash_{\mathsf{S}+t(\Sigma)} t(\varphi_1 \wedge ... \wedge \varphi_n) \to \bot$. But then also $\vdash_{\mathsf{D}+\Sigma} t(\varphi_1 \wedge ... \wedge \varphi_n) \to \bot$ (using lemma 18 on formulas in Σ) and by lemma 18, $\vdash_{\mathsf{D}+\Sigma} \varphi_1 \wedge ... \wedge \varphi_n \to \bot$, which is a contradiction to Γ 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 Γ 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 $\varphi \in \Sigma$. □

4. Terminating tableau systems

In this section we will add to the meta-theory of our 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 will start by providing a tableau system for the static logic SSNL in Section 4.1 before moving on to a tableau system for the dynamic logic DSNL.

4.1. A tableau system for SSNL

Let us start by providing the tableau rules of the tableau system for SSNL and explain how tableau proofs work in general. The approach is highly inspired by the work of Bolander and Blackburn [18, 22]. In fact, the tableau system is identical to theirs with the exception of rules to deal with our special feature propositions. The proof of termination, soundness, and completeness are also more or less identical to the proofs in [18] and thus, they are left out of this section. However, to make the paper self-contained, as the proofs are reused in the next section where we consider the full dynamic logic DSNL, we have included all the proofs in Appendix B.

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

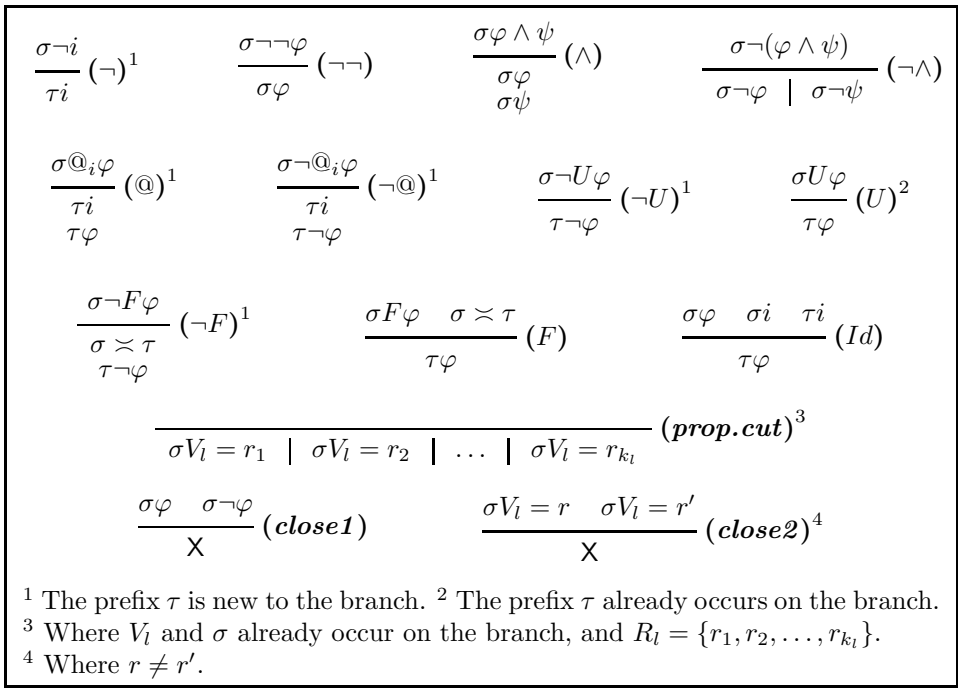

Figure 4: Tableau rules for the logic SSNL.

We assume a new countable infinite set of *prefixes* Pref. We will normally denote elements of Pref by σ, τ, ρ... and so on. The formulas labeling notes in the tableaux will be *prefixed formulas* of the form $\sigma\varphi$, for a $\sigma \in$ Pref and φ a $\mathcal{L}_{\mathcal{SSN}}$-formula, or *accessibility formulas* of the form $\sigma \asymp \tau$ for $\sigma, \tau \in$ Pref.[9]

The tableau rules for the tableau system for SSNL are given in Figure 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 note(s) labeled by the formula(s) below the line. If more than one formula occur above the line, all these formulas have to occur on a branch before the rule can be applied. If several formulas occurs below the horizontal line separated by vertical lines, this means that the branch is split into several new branches each expanded with a note labeled by the given formula. Ignoring the accessibility formulas and formulas of the form σi, 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.

[9]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 (see Appendix B) 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.

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 constrains that no prefix generation rule is applied twice to the same premise on the same branch and a formula is never added to the branch if it already occurs on it. If one of the rules (*close*1) or (*close*2) has been applied to a branch, no other rules can be applied to that branch. A branch of a tableau is called closed if one of the rules (*close*1) or (*close*2) have been applied to it, otherwise the branch 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 φ is a closed tableau with $\sigma\neg\varphi$ as the root formula (for some prefix σ).

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 possible worlds 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 relation has 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 [18]. To define this mechanism we first need some definitions:

Definition 20. *For a branch Θ and a prefix σ occurring on Θ, we define the set:*

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

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

Now, in addition to the already mentioned constrains 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 σ is a urfather on that branch.*

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

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

Now, given a formula φ 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 φ is valid, or the tableau will be open, in which case the model construction of the completeness proof (see Appendix B) provides us with a counter-model to the validity of φ. Thus, we have a decision procedure for the logic SSNL, and it follows that:

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

The completeness of the tableau system is, as previously mentioned, shown in Appendix B. We here state the completeness theorem explicitly though:

Theorem 4 (Completeness for SSNL)**.** *If φ is valid in* SSNL*, then there is a tableau proof of φ.*

An interesting feature of the completeness proofs of tableau systems, as the one presented in Appendix B, is that they provide a way of constructing models satisfying particular formulas. This technique can be used for more than just showing completeness as we be discussed in Section 5.

4.2. A tableau system for DSNL

We extend the tableau system of Section 4.1 to a tableau system for DSNL. 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 [19]. Again, we will start by providing the tableau rules, then we will show that the tableau system is terminating before, finally, moving on to the issues of soundness and completeness.

Before introducing the tableau system, it should be remarked that due to the translation of DSNL into SSNL, and the sound, complete, and terminating tableau system for SSNL just given, we could derive a decision procedure for DSNL by simply first translating any $\mathcal{L}_{\mathcal{DSN}}$ formula we want to check for validity into $\mathcal{L}_{\mathcal{SSN}}$ and then use the decision procedure for SSNL just specified. However, as argued for dynamic epistemic logics in [19], when searching for actual tableau proof it can be an advantage to have a direct tableau system for DSNL instead of always going through the translation into SSNL.

Before giving the tableau rules for DSNL, we define a "one step translation" of $\mathcal{L}_{\mathcal{DSN}}$ formulas of the form $[\mathcal{D}]\varphi$.

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

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

The rules for the tableau system for DSNL are shown in Figure 5. What we have done is essentially translate the reduction axioms of Figure 2 into tableau rules and add them to the tableau rules for SSNL. In this way, we are translating formulas "on-the-fly" and only when needed. As mentioned, this is inspired by [19].

We will put the same constraints on constructing tableaux as described in Section 4.1, including the loop-check condition.

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

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

Figure 5: Tableau rules for the logic DSNL.

We will now consider termination. The termination proof will follow the structure of the proof of termination for the tableau system for SSNL, which was sketched in Section 4.1 and given in details in Appendix B. However, we will need a new notion of what a subformula is to ensure that all formulas occurring in a tableau are subformulas of the root formula. The complexity measure c, defined in Definition 16, plays an important role in this new notion of subformula, which we thus name *c-subformula*:

Definition 23 (c-subformula)**.** *A formula ψ is said to be a* c-subformula *of a formula φ if the following are satisfied:*

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

With this new 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 district formulas φ 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 φ 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, as well. Now, allowing these finitely many dynamic transformation modalities $[\mathcal{D}]$ to occur, again, only finitely many formulas φ with $c(\varphi) \leq k$ can be constructed. Denote this set of formulas by $\mathcal{A}_1$. These formulas can now be used to define a new finite set of dynamics transformation $\mathcal{D} = (\Phi, \mathsf{post})$ with $\Phi \subseteq \mathcal{A}_1$. The recursive definition of $\mathcal{A}_m$ could easily continue this way 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 dynamics transformation satisfying the requirement and thus, for some $m \in \mathbb{N}$, $\mathcal{A}_m = \mathcal{A}_{m+1} = \mathcal{A}_{m+2} \ldots$. Hence, the set

$$\bigcup_{m \in \mathbb{N}} \mathcal{A}_m$$

is a finite union of finite sets and is thus itself finite. Moreover, this set contains all formulas φ with $c(\varphi) \leq k$. □

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

Proof. Let φ be a $\mathcal{L}_{\mathcal{DSN}}$ formula. We might have infinitely many nominals in our language, but by the second requirement in Definition 23 there can only be finitely many of them occurring in the c-subformulas of φ. Hence, by Lemma 24 we can only construct finitely many formulas ψ with these nominals satisfying $c(\psi) \leq c(\varphi)$. Thus, the set of c-subformulas of φ must be finite. □

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 Figure 4 and Figure 5. For the rules $(\mathcal{D})$ and $(\neg\mathcal{D})$ note that it follows by similar reasoning as in the proof of 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 $\mathcal{T}$, φ is a c-subformula of φ_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 φ_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 that or equal to $c(\varphi_0)$ since φ_0 is a formula. For these reasons, it follows that a conclusion of any rule application is a c-subformula of φ_0 provided that the premise is a c-subformula of φ_0. Thus, it follows by induction on the number of rule applications in $\mathcal{T}$ that for all formulas $\sigma\varphi$ on $\mathcal{T}$, φ is a c-subformula of φ_0. □

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

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

Having ensured Lemma 28, we can now prove (see Appendix B) Lemma 34 for the tableau system of DSNL, in exactly the same way. However, first we need Definition 33:

Definition 33. *Let Θ be a branch. If a prefix τ has been introduced to the branch using a prefix generation rule on a formula of the form $\sigma\varphi$ we say that τ is generated by σ and write $\sigma \prec_{\Theta} \tau$.*

Lemma 34. *Let Θ be a tableau branch (constructed using the tableau system for* DSNL*). Then Θ is infinite if and only if there exists an infinite chain of prefixes:*

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

Finally, we prove a proposition for DSNL similar to Proposition 1.

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

Proof. With Lemma 34, Lemma 25, and Lemma 27 ensured for the tableau system of DSNL, we can straight-out adopt the proof of Proposition 1 from Appendix B, by replacing the notion of "quasi-subformula" by our new notion of c-subformula. □

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

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

Let us move on to soundness and completeness of the tableau system for DSNL. Soundness follows from the soundness of the tableau system for SSNL (shortly described in Appendix B) 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 SSNL (in Appendix B). The canonical model is constructed in the exact same way. We just need to add two extra cases in the proof of the truth lemma (Lemma 36) for "$\varphi = [\mathcal{D}]\psi$" and "$\varphi = \neg[\mathcal{D}]\psi$" and do the proof by induction on the new complexity measure c.

Expanded proof of Lemma 36. The case $\varphi = [\mathcal{D}]\psi$. Assume that $\sigma[\mathcal{D}]\psi$ occurs on Θ. Then by closure under the ($\mathcal{D}$) rule, $\sigma T([\mathcal{D}]\psi)$ also occurs on Θ. 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 DSNL:

Theorem 6 (Completeness for DSNL)**.** *If φ is valid in* DSNL*, then there is a tableau proof of φ.*

Proof. Just as the proof of Theorem 4 (see Appendix B). □

5. Examples of Applications

In this section, we will provide a few examples of the kind of modeling and reasoning about changes of distribution of features within social networks which DSNL allows for.

5.1. Pluralistic ignorance

Pluralistic ignorance [23] is a phenomenon from social psychology which has been defined in various ways (see in [24, 25, 26, 23, 27]). We will stick to the definition from [27]: *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 the negation of it).

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 also include drinking habits among college students, attitudes towards norms of racial segregation, and many more [28, 29, 30, 31].

In [12], 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 [11]. 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 continue 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 from the (one-property) influence operator assumed in [11], 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 $\mathcal{D}_E = (\Phi_E, \mathsf{post}_E)$:

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

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

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 of our language:

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

Now apply the transformation $\mathcal{D}_E$. Having another look at the preconditions set Φ_E, note that none of them is satisfied at any agent. Therefore, none of the agents will change her behavior (her "expressed belief" state). 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 $\mathsf{post}_E\big((V_I = n \wedge (\langle F\rangle V_E = n \vee [F]V_E = u)) \vee [F]V_E = n\big)(V_E) = n$, 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 $\mathsf{post}_E\big((V_I = b \wedge (\langle F \rangle V_E = b \vee [F]V_E = u)) \vee [F]V_E = b\big)(V_E) = b$, 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 [12] that this will depend on the network structure itself: if the network graph does not contain any 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![10]

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. The classical example is the diffusion of hybrid seed corn among Iowa farmers [32] (–additionally, see the references in [33]). Still, [33] provides some new insights about how social structures affect the spread of microfinance loans in small Indian villages: detailed data on various types of social ties and structures in 43 rural villages in Southern India was collected before a microfinance institution entered the villages. Based on information from the microfinance institution, [33] then compared the data on social networks to the actual diffusion of microfinance loans in the villages.

It is argued in [33] 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 [33], 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 pro-

[10] For more detail about the discussion of the dissolution of pluralistic ignorance, a proof of the claim of stabilization and an upper bound on such stabilization, we refer the reader to [12].

cesses 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 a dynamics.

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 [33] 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, [33] 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

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

Concerning the diffusion of participation, [33] 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

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

With this 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:

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

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

1.	$\sigma_0 \neg\big((@_j\langle F\rangle i \wedge @_i V_P = y) \rightarrow [\mathcal{D}_I]@_j V_I = y\big)$	
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.	$\sigma_1 \neg\big(\big(\langle F\rangle V_P = y \vee FV_I = y\big) \vee\big(\neg(\langle F\rangle V_p = y \vee FV_I = y) \wedge V_I = y\big)\big)$	$(\neg\mathcal{D})$ on 7.
9.	$\sigma_1 \neg(\langle F\rangle V_P = y \vee FV_I = y)$	$(\neg\neg)$ and $(\wedge)$ on 8.
10.	$\sigma_1 \neg\big(\neg(\langle F\rangle V_P = y \vee FV_I = y) \wedge V_I = y\big)$	$(\neg\neg)$ 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.

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

Following this, one can show that after an additional step if 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 [33], 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 say 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. However, our current logic cannot capture this. In the next section we will shortly discuss extensions of our framework to capture such thresholds preconditions and other interesting traits.

6. Conclusion and further research

The main goal of the present paper was to offer a general framework to reason about the 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 from 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 [7, 8, 11]. 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 [11] within our simpler one-dimensional logic, since in [11] 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 [11] 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 [11].

Note that another simplification of the very same two-dimensional hybrid framework of the seminal [7] has been proposed by [9] already. However, while [9] keeps the multi-agent possible worlds epistemic structure and let 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 [7, 11] can be formally encoded by some atomic propositions. By doing this, we gain in generality: DSNL 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 loose 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 defined both in [10] and partially in [13]. While [10] models channel-based communication, [13] investigates how agents in a social network structure have to communicate in order to merge their beliefs, in the way defined by [34] 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 [35], information via interaction structures [36, 37], communication channels in Interpreted Systems [38], and communication channels in a combination of Interpreted Systems and Dynamic Epistemic Logic [39].

6.2. Future Research

This paper is a first step towards showing how a simple modal logic could be used to represent a large class of dynamic processes, such as diffusion, widely studied in network science. However, our logic has limitations. There are several possible extensions that will increase its usefulness for reasoning about even larger classes of models from network science. We will briefly mention such possible extensions, leaving the details for future work.

First of all, in line with [8], one could add the transitive closure operator F^* of the modality F, with the following semantics:

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

where $\asymp^*$ is the transitive closure of the relation $\asymp$.[11] This "community modality" quantifies over what [8] names an agent's "community", that is, the agent's friends, the agent's friends' friends, the agent's friends' friends' friends, ... etc. With such a modality one can quantify over all agents in a strongly connected component and even express that a network is strongly connected [41]. We have left out this modality in our current logic because it is not relevant to the main ideas behind our logic. 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 in the long run. 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:

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

In Section 2, we discussed how to describe stability, but our language as such

[11] Such transitive closure modalities are also fairly standard in Propositional Dynamic Logic (PDL) [40].

cannot capture stabilization. However, 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, [7] gives a characterization of stable and stabilizing models for the particular transformation under consideration, while [12] reduces stabilization of some type of network models to the existence of an odd cycle in the underlying network structure.[12] We have chosen not to include $\langle \mathcal{D}^* \rangle$ in our logic as it is not essential to the ideas we wish to convey. Moreover, adding $\langle \mathcal{D}^* \rangle$ is again likely to complicate the axiomatization and tableau system of our logic – a similar transitive closure operator of Public Announcement Logic actually leads to an undecidable logic [42].

Another line of extensions concerns more fine-grained thresholds when modeling diffusion phenomena such as the microfinance example of the previous section. In [7, 11, 8] the changes considered depend exclusively on whether "all" or "some" of of an agent's neighbors believe/prefer/know something. Similarly, we restrict our notion of "threshold influence" to those thresholds which are definable using our language $\mathcal{L}_{\mathcal{DSN}}$. However, one can argue that many diffusion phenomena involve numerical thresholds. 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 [3, Ch. 19].

To capture these ideas, one could add *counting modalities* "n of my neighbors" (for any $n \in \mathbb{N}$) or *proportional modalities* like "most of my neighbors" (or any other given proportion of them). Formally, for any $n \in \mathbb{N}$, we could add a modality $[\geq nF]$ with the semantics:

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

where $|B|$ denotes the cardinality of the set B. Alternatively we could also add proportional modalities of the form $[\geq \frac{p}{q}F]$ for $p, q \in \mathbb{N}$ with $p \leq q$, with the following semantics:

$$\mathcal{M}, a \models [\geq \tfrac{p}{q}F]\varphi \quad \text{iff} \quad \frac{|\{b \in A \mid b \asymp a \text{ and } \mathcal{M}, b \models \varphi\}|}{|\{b \in A \mid b \asymp a\}|} \geq \frac{p}{q} \ .$$

In addition to the already mentioned example of microfinance, the extended logic could be used to reason about several standard network analysis issues.

[12] Talking about limiting behavior of social network dynamics using transitive closure modalities is also done in [15] for a particular model of opinion dynamics in social networks. However, the framework of [15] differs considerably from the present framework as it is based on a Fuzzy Logic.

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 [3] for a presentation of a theorem without the use of logic and [14] for the same result using logical tools considerably different from our logic).

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 state 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.

Acknowledgments.

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 [12], on which the present paper partially builds. The research of Zoé Christoff leading to these results has received funding from the European Research Council under the European Communitys Seventh Framework Programme (FP7/2007-2013)/ERC Grant agreement no. 283963. 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] Newman, M.E.J.: Networks: An Introduction. Oxford University Press (2010)

[2] Jackson, M.O.: Social and Economic Networks. Princeton University Press (2010)

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

[4] van Benthem, J.: Logical Dynamics of Information and Interaction. Cambridge University Press, The Netherlands (2011)

[5] Baltag, A., Moss, L.S., Solecki, S.: The logic of public announcements, common knowledge and private suspicious. Technical Report SEN-R9922, CWI, Amsterdam (1999)

[6] van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Syntese Library volume 337. Springer, The Netherlands (2008)

[7] Seligman, J., Liu, F., Girard, P.: Logic in the community. In Banerjee, M., Seth, A., eds.: Logic and Its Applications. Volume 6521 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2011) 178–188

[8] Zhen, L., Seligman, J.: A logical model of the dynamics of peer pressure. Electronic Notes in Theoretical Computer Science **278**(0) (2011) 275 – 288

[9] Ruan, J., Thielscher, M.: A logic for knowledge flow in social networks. In Wang, D., Reynolds, M., eds.: AI 2011: Advances in Artificial Intelligence. Volume 7106 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2011) 511–520

[10] Sano, K., Tojo, S.: Dynamic epistemic logic for channel-based agent communication. In Lodaya, K., ed.: Logic and Its Applications. Volume 7750 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 109–120

[11] Liu, F., Seligman, J., Girard, P.: Logical dynamics of belief change in the community. Synthese (2014) 1–29

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

[13] Christoff, Z.: A logic for social influence through communication. In Lorini, E., ed.: Proceedings of Eleventh European Workshop on Multi-Agent Systems (EUMAS 2013). Volume 1113 of CEUR. (2013) 31–39

[14] Christoff, Z., Rendsvig, R.: Dynamic logics for threshold models and their epistemic extension. (2014)

[15] Hansen, J.U.: 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 27–30 (2014, to appear)

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

[17] Blackburn, P., ten Cate, B.: Pure extensions, proof rules, and hybrid axiomatics. Studia Logica **84**(2) (2006) 277–322

[18] Bolander, T., Blackburn, P.: Termination for hybrid tableaus. Journal of Logic and Computation **17**(3) (2007) 517–554

[19] Hansen, J.U.: Terminating tableaux for dynamic epistemic logics. Electronic Notes in Theoretical Computer Science **262** (2010) 141–156 Proceedings of the 6th workshop on Methods for Modalities (M4M-6 2009).

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

[21] Wang, Y., Cao, Q.: On axiomatizations of public announcement logic. Synthese **190**(1) (2013) 103–134

[22] Bolander, T., Blackburn, P.: Terminating tableau calculi for hybrid logics extending K. Electronic Notes in Theoretical Computer Science **231** (2009) 21–39 Proceedings of the 5th workshop on Methods for Modalities (M4M-5 2007).

[23] O'Gorman, H.J.: The discovery of pluralistic ignorance: An ironic lesson. Journal of the History of the Behavioral Sciences **22**(October) (1986) 333–347

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

[25] Halbesleben, J.R.B., Buckley, M.R.: Pluralistic ignorance: historical development and organizational applications. Management Decision **42**(1) (2004) 126–138

[26] Miller, D.T., McFarland, C.: When social comparison goes awry: The case of pluralistic ignorance. In Suls, J., Wills, T., eds.: Social comparison: Contemporary theory and research. Erlbaum, Hillsdale, NJ (1991) 287–313

[27] Bjerring, J.C., Hansen, J.U., Pedersen, N.J.L.L.: On the rationality of pluralistic ignorance. Synthese (2014) 1–26

[28] Miller, D.T., McFarland, C.: Pluralistic ignorance; when similarity is interpreted as dissimilarity. Journal of Personality and Social Psychology **53** (1987) 298–305

[29] Prentice, D.A., Miller, D.T.: Pluralistic ignorance and alcohol use on campus: Some consequences of misperceiving the social norm. Journal of Personality and Social Psychology **64**(2) (1993) 243–256

[30] Fields, J.M., Schuman, H.: Public beliefs about the beliefs of the public. The Public Opinion Quarterly **40**(4) (1976) 427–448

[31] O'Gorman, H.J., Garry, S.L.: Pluralistic ignorance – a replication and extension. The Public Opinion Quarterly **40**(4) (1976) 449–458

[32] Ryan, B., Gross, N.C.: The diffusion of hybrid seed corn in two iowa communities. Rural sociology **8**(1) (1943) 15–24

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

[34] Baltag, A., Smets, S.: Protocols for belief merge: Reaching agreement via communication. Logic Journal of the IGPL **21**(3) (2013) 468–487

[35] Roelofsen, F.: Exploring logical perspectives on distributed information and its dynamics. Master's thesis, University of Amsterdam (2005)

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

[37] Apt, K.R., Witzel, A., Zvesper, J.A.: 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) 4–13

[38] Parikh, R., Ramanujam, R.: A knowledge based semantics of messages. Journal of Logic, Language and Information **12**(4) (2003) 453–467

[39] Wang, Y., Sietsma, F., van Eijck, J.: Logic of information flow on communication channels. In Grossi, D., Kurzen, L., Velzques-Queseda, F.R., eds.: Logic and Interactive Rationality, Yearbook 2009, Amsterdam, ILLC (2010)

[40] Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. The MIT Press, Cambridge (2000)

[41] Gate, J., Stewart, I.A.: The expressibility of fragments of hybrid graph logic on finite digraphs. Journal of Applied Logic **11**(3) (2013) 272–288

[42] Miller, J.S., Moss, L.S.: The undecidability of iterated modal relativization. Studia Logica **79**(3) (2005) 373–407

Appendix A: Proofs of Section 3.2

In this appendix we provide the missing proofs of Section 3.2.

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

$$(\mathcal{M}^{\mathcal{D}})^{\mathcal{D}'} = \mathcal{M}^{(\mathcal{D};\mathcal{D}')} \tag{13}$$

Proof. Let $\mathcal{M} = (A, \asymp, g, \nu)$ be an arbitrary network model and let $\mathcal{D} = (\Phi, \mathsf{post})$ and $\mathcal{D}' = (\Phi', \mathsf{post}')$ be two dynamic transformations. First note that $(\mathcal{D}; \mathcal{D}')$ is always defined. Moreover, from Definition 8 it follows that we can write $\mathcal{M}^{\mathcal{D}} = (A, \asymp, g, \nu')$, $(\mathcal{M}^{\mathcal{D}})^{\mathcal{D}'} = (A, \asymp, g, \nu_1)$ and $\mathcal{M}^{(\mathcal{D};\mathcal{D}')} = (A, \asymp, g, \nu_2)$. Hence, to show the equality (13) we only need to show that $\nu_1 = \nu_2$. That is we need to show that for all $a \in A$ and all $l \in \{1, ..., n\}$:

$$\nu_1(a)(l) = \nu_2(a)(l). \tag{14}$$

Now, let $a \in A$ and $l \in \{1, ..., n\}$ be given. We first distinguish two cases: (a) there is $\varphi \in \Phi$ such that $\mathcal{M}, a \models \varphi$ and $l \in dom(\mathsf{post}(\varphi))$, and (b) there is no such φ.

Consider the case (a). In this case, by Definition 8, we have that $\nu'(a)(l) = \mathsf{post}(\varphi)(l)$. Now consider the following two sub-cases: (a.i) there is a $\psi \in \Phi'$ such that $\mathcal{M}^{\mathcal{D}}, a \models \psi$ and $l \in dom(\mathsf{post}'(\psi)$, and (a.ii) there is no such ψ.

Consider the sub-case (a.i). In this case, according to Definition 8, $\nu_1(a)(l) = \mathsf{post}'(\psi)(l)$. Note also that now $\varphi \wedge [\mathcal{D}]\psi$ is satisfiable at a in $\mathcal{M}$ and hence, by Definiton 8 and Definition 13

$$\nu_2(a)(l) = \mathsf{post}''(\varphi \wedge [\mathcal{D}]\psi)(l) = \mathsf{post}'(\psi)(l) = \nu_1(a)(l).$$

Consider the sub-case (a.ii). By Definition 8 we have that $\nu_1(a)(l) = \mathsf{post}(\varphi)(l)$ and by Definition 13 (the second or third case for post'') it follows that $\nu_2(a)(l) = \mathsf{post}(\varphi)(l)$, as well.

Now consider the case (b). In this case, by Definition 8, we have that $\nu'(a)(l) = \nu(a)(l)$. Consider again the following two sub-cases: (b.i) there is a $\psi \in \Phi'$ such that $\mathcal{M}^{\mathcal{D}}, a \models \psi$ and $l \in dom(\mathsf{post}'(\psi)$, and (b.ii) there is no such ψ.

Consider the sub-case (b.i). In this case, by Definition 8, we have that $\nu_1(a)(l) = \mathsf{post}'(\psi)(l)$. On the other hand, by Definition 13 (the first or fourth case for post'') it follows that $\nu_2(a)(l) = \mathsf{post}'(\psi)(l)$, as well.

Consider finally the sub-case (b.ii). From Definition 8 it follows that $\nu_1(a)(l) = \nu(a)(l)$. Since there are no $\varphi \in \Phi$ or $\psi \in \Phi'$ such that $l \in dom(\mathsf{post}(\varphi))$ or $l \in dom(\mathsf{post}'(\psi))$ clearly from Definition 13, l is not in $dom(\mathsf{post}''(\chi))$ for any $\chi \in \Phi''$. Hence, by Definition 8, $\nu_2(a)(l) = \nu(a)(l)$, as well. This completes the proof. □

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

$$\mathcal{M}, a \models [\mathcal{D}]V_l{=}r \quad \textit{iff} \quad \mathcal{M}, a \models \Big(\bigvee_{\varphi\in\Phi,\ \mathsf{post}(\varphi)(l)=r} \varphi\Big) \vee \Big(\neg\big(\bigvee_{\varphi\in\Phi,\ l\in dom(\mathsf{post}(\varphi))} \varphi\big) \wedge V_l{=}r\Big) \tag{15}$$

$$\mathcal{M}, a \models [\mathcal{D}]i \quad \textit{iff} \quad \mathcal{M}, a \models i \tag{16}$$

$$\mathcal{M}, a \models [\mathcal{D}]\neg\varphi \quad \textit{iff} \quad \mathcal{M}, a \models \neg[\mathcal{D}]\varphi \tag{17}$$

$$\mathcal{M}, a \models [\mathcal{D}](\varphi \wedge \psi) \quad \textit{iff} \quad \mathcal{M}, a \models [\mathcal{D}]\varphi \wedge [\mathcal{D}]\psi \tag{18}$$

$$\mathcal{M}, a \models [\mathcal{D}]@_i\varphi \quad \textit{iff} \quad \mathcal{M}, a \models @_i[\mathcal{D}]\varphi \tag{19}$$

$$\mathcal{M}, a \models [\mathcal{D}]F\varphi \quad \textit{iff} \quad \mathcal{M}, a \models F[\mathcal{D}]\varphi \tag{20}$$

$$\mathcal{M}, a \models [\mathcal{D}]U\varphi \quad \textit{iff} \quad \mathcal{M}, a \models U[\mathcal{D}]\varphi \tag{21}$$

$$\mathcal{M}, a \models [\mathcal{D}][\mathcal{D}']\varphi \quad \textit{iff} \quad \mathcal{M}, a \models [(\mathcal{D};\mathcal{D}')]\varphi \tag{22}$$

Proof. Proof of (15). This was provided in Section 3.2.

Proof of (16). Note that $\mathcal{M}, a \models [\mathcal{D}]i$ if, and only if, $\mathcal{M}^{\mathcal{D}}, a \models i$. But since the function g assigning agents to nominals does not change under the update with $\mathcal{D}$, $\mathcal{M}^{\mathcal{D}}, a \models i$ if, and only if, $\mathcal{M}, a \models i$

Proof of (17)-(22). The proofs of $(17) - (22)$ are very similar, so we only do the cases (17), (20), and (22).

Proof of (17). We have the following equivalences:

$$\begin{array}{lll} \mathcal{M}, a \models [\mathcal{D}]\neg\varphi & \text{iff} & \mathcal{M}^{\mathcal{D}}, a \models \neg\varphi \\ & \text{iff} & \mathcal{M}^{\mathcal{D}}, a \not\models \varphi \\ & \text{iff} & \mathcal{M}, a \not\models [\mathcal{D}]\varphi \\ & \text{iff} & \mathcal{M}, a \models \neg[\mathcal{D}]\varphi. \end{array}$$

Proof of (20). We have the following equivalences:

$$\begin{array}{lll} \mathcal{M}, a \models [\mathcal{D}]F\varphi & \text{iff} & \mathcal{M}^{\mathcal{D}}, a \models F\varphi \\ & \text{iff} & \mathcal{M}^{\mathcal{D}}, b \models \varphi, \quad \text{for all } b \in A \text{ with } a \asymp b \\ & \text{iff} & \mathcal{M}, b \models [\mathcal{D}]\varphi, \quad \text{for all } b \in A \text{ with } a \asymp b \\ & \text{iff} & \mathcal{M}, a \models F[\mathcal{D}]\varphi. \end{array}$$

Proof of (22). By Lemma 14, we have the following equivalences:

$$\begin{array}{lll} \mathcal{M}, a \models [\mathcal{D}][\mathcal{D}']\varphi & \text{iff} & \mathcal{M}^{\mathcal{D}}, a \models [\mathcal{D}']\varphi \\ & \text{iff} & (\mathcal{M}^{\mathcal{D}})^{\mathcal{D}'}, a \models \varphi \\ & \text{iff} & \mathcal{M}^{(\mathcal{D};\mathcal{D}')}, a \models \varphi \\ & \text{iff} & \mathcal{M}, a \models [(\mathcal{D};\mathcal{D}')]\varphi. \end{array}$$

□

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. $c([\mathcal{D}]V_l = r) > c\big(\big(\bigvee_{\varphi\in\Phi,\mathsf{post}(\varphi)(l)=r}\varphi\big) \vee \big(\neg(\bigvee_{\varphi\in\Phi,\ l\in dom(\mathsf{post}(\varphi))}\varphi) \wedge V_l{=}r\big)\big)$
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. $c([\mathcal{D}][\mathcal{D}']\varphi) > c([\mathcal{D}; \mathcal{D}']\varphi)$

Proof. Let $\mathcal{D} = (\Phi, \mathsf{post})$, $K = 3 \cdot |\Phi| + 3$, and $|\Phi| = n$.

1. It suffices to note that $c(i) = 1$, while $c([\mathcal{D}]i) = K + c(\mathcal{D})$.
2. Since $c(V_l = r) = 1$,
   $c([\mathcal{D}]V_l = r) = K + c(\mathcal{D}) = 3n + 3 + c(\mathcal{D})$
   It is easy to check that:
   a) $c(\bigvee_{\psi\in\Phi}\psi) \leq 3n - 3 + max\{c(\psi) \mid \psi \in \Phi\}$
      Now note that $|\{\psi \in \Phi \mid \mathsf{post}(\psi)(l) = r\}| \leq n$ and therefore:
      $c(\bigvee_{\psi\in\Phi,\mathsf{post}(\psi)(l)=r}\psi) \leq 3n - 3 + max\{c(\psi) \mid \psi \in \Phi\}$
   b) $c(\neg(\bigvee_{\psi\in\Phi\psi}\psi) \wedge V_l{=}r) \leq 3n - 1 + max\{c(\psi) \mid \psi \in \Phi\}$
      Again, note that $|\{\psi \in \Phi \mid l \in dom(\mathsf{post}(\psi))\}| \leq n$ and therefore:
      $c\big(\neg(\bigvee_{\psi\in\Phi,\ l\in dom(\mathsf{post}(\psi))}\psi) \wedge V_l{=}r\big) \leq 3n - 1 + max\{c(\psi) \mid \psi \in \Phi\}$
   c) $c\big(\big(\bigvee_{\varphi\in\Phi,\mathsf{post}(\varphi)(l)=r}\varphi\big) \vee \big(\neg(\bigvee_{\varphi\in\Phi,\ l\in dom(\mathsf{post}(\varphi))}\varphi) \wedge V_l{=}r\big)\big) =$
      $3 + max\big(c(\bigvee_{\psi\in\Phi,\mathsf{post}(\psi)(l)=r}\psi), c\big(\neg(\bigvee_{\psi\in\Phi,\ l\in dom(\mathsf{post}(\psi))}\psi) \wedge V_l{=}r\big)\big)$
   d) It follows from the above that:
      $c\big(\big(\bigvee_{\varphi\in\Phi,\mathsf{post}(\varphi)(l)=r}\varphi\big) \vee \big(\neg(\bigvee_{\varphi\in\Phi,\ l\in dom(\mathsf{post}(\varphi))}\varphi) \wedge V_l{=}r\big)\big) \leq$
      $3n + 2 + max\{c(\psi) \mid \psi \in \Phi\} =$
      $3n + 2 + c(\mathcal{D})$
3. For the case of the negation, it is easy to compare:
   a) $c([\mathcal{D}]\neg\varphi) = (K + c(\mathcal{D})) \cdot c(\neg\varphi) = (K + c(\mathcal{D})) \cdot (1 + c(\varphi)) =$
      $K + c(\mathcal{D}) + K \cdot c(\varphi) + c(\mathcal{D}) \cdot c(\varphi)$
   b) $c(\neg[\mathcal{D}]\varphi) = 1 + (K + c(\mathcal{D})) \cdot c(\varphi) =$
      $1 + K \cdot c(\varphi) + c(\mathcal{D}) \cdot c(\varphi)$
4. The case of $\Box$ is entirely similar to the case of negation:
   a) $c([\mathcal{D}]\Box\varphi) = K + c(\mathcal{D}) + K \cdot c(\varphi) + c(\mathcal{D}) \cdot c(\varphi)$
   b) $c(\Box[\mathcal{D}]\varphi) = 1 + K \cdot c(\varphi) + c(\mathcal{D}) \cdot c(\varphi)$
5. For the case of conjunction:
   a) $c([\mathcal{D}](\varphi \wedge \psi)) =$
      $(K + c(\mathcal{D})) \cdot c(\varphi \wedge \psi) =$
      $(K + c(\mathcal{D})) \cdot (1 + max(c(\varphi), c(\psi))) =$
      $K + c(\mathcal{D}) + (K + c(\mathcal{D})) \cdot max(c(\varphi), c(\psi))$

b) $c([\mathcal{D}]\varphi \wedge [\mathcal{D}]\psi) =$
$1 + max(c([\mathcal{D}]\varphi), c([\mathcal{D}]\psi)) =$
$1 + max((K + c(\mathcal{D})) \cdot c(\varphi), (K + c(\mathcal{D})) \cdot c(\psi)) =$
$1 + (K + c(\mathcal{D})) \cdot max(c(\varphi), c(\psi))$

6. Let $\mathcal{D}' = (\Phi', \mathsf{post}')$, $K' = 3 \cdot |\Phi'| + 3$, and $|\Phi'| = n'$. Moreover, let $\mathcal{D}; \mathcal{D}' = (\Phi'', \mathsf{post}'')$, $K'' = 3 \cdot |\Phi''| + 3$, and $|\Phi''| = n''$ for $\mathcal{D}; \mathcal{D}'$ given by Definition 13:

   a) $c([\mathcal{D}][\mathcal{D}']\varphi) =$
   $(K + c(\mathcal{D})) \cdot c([\mathcal{D}']\varphi) =$
   $(K + c(\mathcal{D})) \cdot ((K' + c(\mathcal{D}')) \cdot c(\varphi) =$
   $(K \cdot K' + K' \cdot c(\mathcal{D}) + K \cdot c(\mathcal{D}') + c(\mathcal{D}) \cdot c(\mathcal{D}')) \cdot c(\varphi)$

   b) Recall that $c(\mathcal{D}; \mathcal{D}') = max\{c(\chi) \mid \chi \in \Phi''\}$, where, according to Definition 13, each formula $\chi \in \Phi''$ is of one of the following three types:

      i) $\chi = \varphi \wedge [\mathcal{D}]\psi$, for some $\varphi \in \Phi$ and some $\psi \in \Phi'$.
      $c(\varphi \wedge [\mathcal{D}]\psi) =$
      $1 + max(c(\varphi), c([\mathcal{D}]\psi)) =$
      $1 + max(c(\varphi), (K + c(\mathcal{D})) \cdot c(\psi))$
      Since we know that $c(\mathcal{D}) \geq c(\varphi)$ and $c(\mathcal{D}') \geq c(\psi)$,
      we can infer that $c(\chi) \leq 1 + ((K + c(\mathcal{D})) \cdot c(\mathcal{D}'))$.

      ii) $\chi = \varphi \wedge [\mathcal{D}]\big(\bigwedge_{\psi \in \Phi'} \neg\psi\big)$, for some $\varphi \in \Phi$. Let $|\Phi'| = n'$.
      $c(\varphi \wedge [\mathcal{D}]\big(\bigwedge_{\psi \in \Phi'} \neg\psi\big)) =$
      $1 + max(c(\varphi), c([\mathcal{D}](\bigwedge_{\psi \in \Phi'} \neg\psi))) \leq$
      $1 + max(c(\varphi), (K + c(\mathcal{D})) \cdot (n' + max\{c(\psi) \mid \psi \in \Phi'\})) =$
      $1 + max(c(\varphi), (K + c(\mathcal{D})) \cdot (n' + c(\mathcal{D}')))=$
      $1 + (K + c(\mathcal{D})) \cdot (n' + c(\mathcal{D}'))$

      iii) $\chi = \bigwedge_{\varphi \in \Phi} \neg\varphi \wedge [\mathcal{D}]\psi$, for some $\psi \in \Phi'$.
      $c(\bigwedge_{\varphi \in \Phi} \neg\varphi \wedge [\mathcal{D}]\psi) =$
      $1 + max(n + max\{c(\varphi) \mid \varphi \in \Phi\}, (K + c(\mathcal{D})) \cdot c(\psi)) =$
      $1 + max(n + c(\mathcal{D}), (K + c(\mathcal{D})) \cdot c(\psi)) =$
      (since $K > n$)
      $1 + (K + c(\mathcal{D})) \cdot c(\psi)$
      Since $c(\psi) \leq c(\mathcal{D}')$, we can infer that
      $c(\chi) \leq 1 + (K + c(\mathcal{D})) \cdot c(\mathcal{D}')$.

      From i),ii) and iii) we can conclude that:
      $c(\mathcal{D}; \mathcal{D}') \leq 1 + (K + c(\mathcal{D})) \cdot (n' + c(\mathcal{D}'))$ and therefore that:
      $c([\mathcal{D}; \mathcal{D}']\varphi) \leq (K'' + 1 + (K + c(\mathcal{D})) \cdot (n' + c(\mathcal{D}'))) \cdot c(\varphi)$

   c) To terminate the proof, we need to compare $c([\mathcal{D}][\mathcal{D}']\varphi)$ with $c([\mathcal{D}; \mathcal{D}']\varphi)$. Ignoring the complexity of φ, since it is a coefficient in both, we need to compare the following coefficients 1) and 2) below:

      1) $(K + c(\mathcal{D})) \cdot (K' + c(\mathcal{D}')) =$
      $K \cdot K' + K' \cdot c(\mathcal{D}) + K \cdot c(\mathcal{D}') + c(\mathcal{D}) \cdot c(\mathcal{D}')$

2) $K'' + 1 + (K + c(\mathcal{D})) \cdot (n' + c(\mathcal{D}')) =$
$K'' + 1 + K \cdot n' + n' \cdot c(\mathcal{D}) + K \cdot c(\mathcal{D}') + c(\mathcal{D}) \cdot c(\mathcal{D}')$
Simplifying even more, what we want to show is the following:
$K \cdot K' + K' \cdot c(\mathcal{D}) > K'' + 1 + K \cdot n' + n' \cdot c(\mathcal{D})$

Let us now use the fact that, by definition, $K = 3n + 3, K' = 3n' + 3, K'' = 3n'' + 3$. Note that we can also give an upper bound to $n'' = |\Phi''|$. There are at most $n \cdot n'$ formulas $\in \Phi''$ of the above type i), at most n of type ii) and at most n' of type iii). It follows that $n'' \leq n \cdot n' + n + n'$ and therefore: $K'' = 3n'' + 3 \leq 3nn' + 3n + 3n' + 3$.

1) $K \cdot K' + K' \cdot c(\mathcal{D}) =$
$(3n + 3) \cdot (3n' + 3) + (3n' + 3) \cdot c(\mathcal{D}) =$
$9nn' + 9n + 9n' + 9 + 3n' \cdot c(\mathcal{D}) + 3 \cdot c(\mathcal{D})$

2) $K'' + 1 + K \cdot n' + n' \cdot c(\mathcal{D}) \leq$
$(3nn' + 3n + 3n' + 3) + 1 + (3n + 3) \cdot n' + n' \cdot c(\mathcal{D}) =$
$3nn' + 3n + 3n' + 4 + 3nn' + 3n' + n' \cdot c(\mathcal{D}) =$
$6nn' + 3n + 6n' + 4 + n' \cdot c(\mathcal{D})$

Hence, we can conclude that $c([\mathcal{D}][\mathcal{D}']\varphi) > c([\mathcal{D}; \mathcal{D}']\varphi)$.

□

Appendix B: Termination and Completeness of the tableau system for SSNL

In this appendix we prove that the tableau system for SSNL, given by Figure 4 and the constraints mentioned in Section 4.1, is both terminating and complete. The proofs are more or less just adoptions of the proofs given in [18], however, we will need to refer to these proofs when proving the corresponding results for the tableau system for DSNL proven in Section 4.2. Hence, we provide the proofs in details here.

We will start by showing termination. All tableaux referred to in this appendix will tableaux constructed using the tableau system for SSNL.

6.3. Termination of the tableau system for SSNL

We first prove, a subformula property that is essential for termination of the tableau system. To state and prove this we need the following definition:

Definition 29 (Quasi-subformula)**.** *For feature propositions, we make the convention that $V_l = r$ is a subformula of $V_l = r'$ for all $r, r' \in R_l$. Then, a formula φ is said to be a* quasi-subformula *of ψ if φ is a subformula of ψ, or if φ has the form $\neg\chi$ and χ is a subformula of ψ.*

Lemma 30. *For every formula φ, the set of quasi-subformulas of φ is finite.*

Proof. This easily follows from Definition 29. □

Lemma 31. *Let $\mathcal{T}$ be a tableau with root formula $\sigma\varphi$. If the formula $\tau\psi$ occurs on $\mathcal{T}$, then ψ is a quasi-subformula of φ.*

Proof. The proof goes by induction on the application of rules. For all rules, it easy to check that if they have quasi-subformulas as premises the conclusion will also be a quasi-subformula. □

From this lemma it easily follows, that

Lemma 32. *For all tableau branches Θ and prefixes σ occurring on Θ, the set $T^{\Theta}(\sigma)$ is finite.*

Proof. Let Θ be a tableau branch with root formula $\sigma_0\varphi_0$ and σ an arbitrary prefix occurring on Θ. Then, from Lemma 31,

$$T^{\Theta}(\sigma) \subseteq \{\varphi \mid \varphi \text{ is a quasi-subformula of } \varphi_0\}.$$

Since Lemma 30 implies that the set on the right is finite, it follows that $T^{\Theta}(\sigma)$ is finite, as well. □

Definition 33. *Let Θ be a branch. If a prefix τ has been introduced to the branch using a prefix generation rule on a formula of the form $\sigma\varphi$ we say that τ is generated by σ and write $\sigma \prec_{\Theta} \tau$.*

In similar manners as in [18], we can now show that:

Lemma 34. *Let Θ be a tableau branch. Then Θ is infinite if and only if there exists an infinite chain of prefixes:*

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

Proof. The "if" direction is obvious. For the other direction, assume that Θ is infinite. Then Θ contains infinitely many distinct prefixes. For, assume towards a contradiction that Θ only contains finitely many prefixes. Then, from Lemma 32 it follows that there are only finitely many prefixed formulas. Furthermore, if there is only finitely many prefixes there can only be finitely many accessibility formulas of the form $\sigma \asymp \tau$. Thus, it follows that Θ is finite, which is a contradiction. Thus, Θ contains infinitely many prefixes.

Now, as proved in Lemma 4.2 of [18], the set of prefixes on Θ together with the relation $\prec_{\Theta}$ constitutes a well-founded, finitely branching tree and since the set of prefixes is infinite, so is this tree. However, then it follows from König's Lemma that the tree have an infinite branch, which exactly is an infinite chain of prefixes:

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

□

We now have what it takes to prove termination of the tableau system:

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

Proof. Assume towards a contradiction that there is an infinite tableau constructed using the tableau system. Since tableaux are finitely branching trees, the infinite tableau will contain an infinite branch Θ and by Lemma 34 an infinite chain of prefixes

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

Now, let

$$A = \{\varphi \mid \varphi \text{ is a quasi-subformula of the root formula}\} \ .$$

Note that A must be finite by Lemma 30. For each $i > 0$, let Θ_i be the initial segment of Θ up to, but not including, the first occurrence of σ_{i+1} on Θ. Now, by Lemma 31 all the sets

$$T^{\Theta_1}(\sigma_1), T^{\Theta_2}(\sigma_2), \ \dots$$

are subset of the set A. Thus, since A is finite, we can find i and j with $i < j$ such that $T^{\Theta_i}(\sigma_i) = T^{\Theta_j}(\sigma_j)$. Since the first occurrence of σ_{i+1} on Θ is before the first occurrence of σ_{j+1} on Θ, Θ_i is an initial segment of Θ_j. Then, clearly $T^{\Theta_i}(\sigma_i) \subseteq T^{\Theta_j}(\sigma_i)$, and we obtain that

$$T^{\Theta_j}(\sigma_j) \subseteq T^{\Theta_j}(\sigma_i).$$

Now, since σ_i is introduced earlier than σ_j on Θ, σ_j cannot be a urfather on Θ_j. By definition, the first formula (on Θ) not on Θ_j is of the form $\sigma_{j+1}\psi$ and because of $\sigma_j \prec_\Theta \sigma_{j+1}$, $\sigma_{j+1}\psi$ must have been introduced on Θ by applying a prefix generating rule to a formula of the form $\sigma_j\varphi$ occurring on Θ_j. However, this contradicts the loop-check constraint as σ_j is not an urfather on Θ_j. Hence, we have reached a contradiction and there cannot be a infinite tableau. □

6.3.1. Soundness and completeness of the tableau system

We now turn to soundness and completeness of the tableau system. Soundness is proved in the standard way for tableau systems and is straight forward by noticing that all the rules of Figure 4 preserve satisfiability (in the same model). This is easy to see and we leave the details for the reader. Soundness now follows by the following short reasoning: If a tableau proof exists for φ (i.e. a closed tableau with $\sigma\neg\varphi$ at the root), then φ has to be valid. If not, $\neg\varphi$ would be satisfiable and since the rules preserve satisfiability all formulas on each of the branches would have to be satisfiable simultaneously for each branch. However, this would contradict that all the branches are closed.

Completeness is also proved in the standard way. That is, from a special kind of open branch we construct a model that will be a model of the root formula. Completeness now follows by the following short reasoning: If φ is valid, there has to be a tableau proof for φ (i.e. a closed tableau with $\sigma\neg\varphi$ at the root). If not, all tableaux with $\sigma\neg\varphi$ at the root would be open. Especially, there would be a open tableau of a special kind with an open branch from which we could

construct a model satisfying $\neg\varphi$, which would contradict that φ is valid.

Now, constructing a model from a special kind of open branch is the key step in the completeness proof and requires substantially more work, to which we now turn. First, we need some more terminology and lemmas. We call a tableau *saturated* if no more rules can be applied to it that satisfy the constraints. We call a branch saturated if it is a branch of a saturated tableau. Thus, saturation of a branch Θ means that Θ is closed under application of every rule except for application of prefix generation rules blocked by the loop-check condition. Note that, due to Proposition 1 we can always extend a tableau to a finite saturated tableau.

Before we can prove completeness we need to prove a few properties of urfathers. Again these are taken from [18].

Lemma 35 (Urfather Lemma)**.** *Let Θ be a saturated branch and let σ and τ be prefixes occurring on Θ. Then the following holds:*

(i) For every formula φ, if $\sigma\varphi$ occurs on Θ, then $u_\Theta(\sigma)\varphi$ will also occur on Θ.

(ii) If there is a nominal i such that both σi and τi occur on Θ, then $u_\Theta(\sigma) = u_\Theta(\tau)$.

(iii) σ is an urfather on Θ if, and only if, $u_\Theta(\sigma) = \sigma$.

Proof. Proof of (i): This follows directly from the definition of an urfather.

Proof of (ii): Since Θ is saturated it is closed under all applications of the (Id) rule. Thus, if σi and τi both occur on Θ then $T^\Theta(\sigma) = T^\Theta(\tau)$. Now, $u_\Theta(\sigma)$ and $u_\Theta(\tau)$ and the earliest introduced prefixes such that $T^\Theta(\sigma) \subseteq T^\Theta(u_\Theta(\sigma))$ and $T^\Theta(\tau) \subseteq T^\Theta(u_\Theta(\tau))$. Hence, $u_\Theta(\sigma)$ and $u_\Theta(\tau)$ must be identical.

Proof of (iii): If $u_\Theta(\sigma) = \sigma$ then clearly σ is an urfather by definition. Now assume that σ is an urfather, i.e. there is a prefix ρ such that $\sigma = u_\Theta(\rho)$. For $u_\Theta(\sigma)$, we have that

$$T^\Theta(\rho) \subseteq T^\Theta(\sigma) \subseteq T^\Theta(u_\Theta(\sigma)).$$

Thus, if $u_\Theta(\sigma)$ was introduced on Θ before σ, it would contradict that σ is the urfather of ρ. But then, since $u_\Theta(\sigma)$ is the earliest introdcued prefix such that $T^\Theta(\sigma) \subseteq T^\Theta(u_\Theta(\sigma))$, $u_\Theta(\sigma) = \sigma$ must be the case. □

As in [18], we can now define a canonical model $\mathcal{M}^\Theta = \langle A^\Theta, \asymp^\Theta, g^\Theta, \nu^\Theta \rangle$ for every open saturated branch Θ in the following way:

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

where σ_0 is just some fixed element of A^Θ.

Let us see why this is well-defined. First, note that $g^\Theta(i)$ is well-defined for all $i \in \mathsf{NOM}$. This is clear if there is no urfather σ such that σi occurs on Θ. If there is σ such that σi occurs on Θ, then this urfather σ is unique by (ii) of Lemma 35. $\nu(\sigma)(l)$ is well-defined for each $\sigma \in A^\Theta$ and $l \in \{1, ..., n\}$. First of all, there is a $r \in R_l$ such that $\sigma V_l = r$ occurs on Θ by the closure under the **(prop.cut)**-rule. Moreover, there cannot be $r, r' \in R_l$ with $r \neq r'$ such that both $\sigma V_l = r$ and $\sigma V_l = r'$ occur on Θ since this would contradict that Θ was open (by the **(close2)**-rule).

The key step is the completeness of the tableau system is the following truth lemma:

Lemma 36 (Truth lemma)**.** *Let Θ be an open saturated branch of the tableau system. For any urfather σ on Θ and any formula φ; if $\sigma\varphi$ occurs on Θ, it holds that $\mathcal{M}^\Theta, \sigma \models \varphi$.*

Proof. The proof goes by induction on the complexity of φ.

The base cases. The cases $\varphi = V_l = r$ and $\varphi = i$ for $V_l = r \in \mathsf{FP}$ and $i \in \mathsf{NOM}$ follow directly from the definition of $\mathcal{M}^\Theta$. For the case φ is of the form $\neg V_l = r$, assume that $\sigma \neg V_l = r$ occurs on Θ. Since Θ is an open branch $\sigma V_l = r$ does not occur on Θ and thus $\nu(\sigma)(l) \neq r$ by definition of ν^Θ. However, this implies that $\mathcal{M}^\Theta, \sigma \models \neg V_l = r$. For the case φ is of the form $\neg i$, assume that $\sigma \neg i$ occurs on Θ. Then clearly σi cannot occur on the open branch Θ and thus, $g^\Theta(i) \neq \sigma$ by the definition of g^Θ. Hence, $\mathcal{M}^\Theta, \sigma \models \neg i$.

The induction cases. The cases where $\varphi = \psi \wedge \chi$, $\varphi = \neg(\psi \wedge \chi)$, or $\varphi = \neg\neg\psi$ are straight forward.

The case $\varphi = F\psi$. Assume that $\sigma F\psi$ occurs on Θ and that $\sigma \asymp^\Theta \tau$. We need to show that $\mathcal{M}^\Theta, \tau \models \psi$. Since, $\sigma \asymp^\Theta \tau$, there is a prefix ρ such that $\tau = u_\Theta(\rho)$ and $\sigma \asymp \rho$ occurs on Θ. By closure under the (F)-rule. $\rho\psi$ must occur on Θ. By (i) of Lemma 35, $\tau\psi$ also occurs on Θ and by the induction hypothesis we get that $\mathcal{M}^\Theta, \tau \models \psi$, as needed.

The case $\varphi = \neg F\psi$. Assume that $\sigma \neg F\psi$ occurs on Θ. Then, since σ is an urfather and Θ is saturated, $\sigma \asymp \tau$ and $\tau\neg\psi$ also occur on Θ for some prefix τ. From $\tau\neg\psi$ being on Θ is follows, by (i) of Lemma 35, that $u_\Theta(\tau)\neg\psi$ occurs on Θ. Furthermore, by the induction hypothesis, it follows that $\mathcal{M}^\Theta, u_\Theta(\tau) \models \neg\psi$. From $\sigma \asymp \tau$ being on Θ is follows that $\sigma \asymp^\Theta u_\Theta(\tau)$ and thus, $\mathcal{M}^\Theta, \sigma \models \neg F\psi$ follows.

The case $\varphi = @_i\psi$. Assume that $\sigma @_i\psi$ occurs on Θ. Since Θ is saturated and σ is an urfather, there is a prefix τ such that τi and $\tau\psi$ both occur on Θ. But then, by (i) of Lemma 35, $u_\Theta(\tau)i$ and $u_\Theta(\tau)\psi$ occur on Θ, as well. Now, by the definition of g^Θ, $g^\Theta(i) = u_\Theta(\tau)$ and by induction, $\mathcal{M}^\Theta, u_\Theta(\tau) \models \psi$. Hence, $\mathcal{M}^\Theta, \sigma \models @_i\psi$.

The case $\varphi = \neg @_i\psi$. Assume that $\sigma \neg @_i\varphi$ occurs on Θ. Since Θ is saturated and σ is an urfather, τi and $\tau\neg\varphi$ occur on Θ, for some prefix τ. Then, by (i) of Lemma 35 $u_\Theta(\tau)i$ and $u_\Theta(\tau)\neg\varphi$ also occur on Θ. By the induction hypothesis this implies that $\mathcal{M}^\Theta, u_\Theta(\tau) \models i$ and $\mathcal{M}^\Theta, u_\Theta(\tau) \models \neg\varphi$. Hence, $\mathcal{M}^\Theta, \sigma \models \neg @_i\varphi$.

The case $\varphi = U\psi$. Assume that $\sigma U\psi$ occurs on Θ and assume that $\tau \in A^{\Theta}$. Then τ occurs on Θ and saturation implies that $\tau\psi$ occurs on Θ, as well. By induction it follows that $\mathcal{M}^{\Theta}, \tau \models \psi$. Since $\tau \in A^{\Theta}$ was arbitrary, $\mathcal{M}^{\Theta}, \tau \models \psi$ must be the case for all $\tau \in A^{\Theta}$. Thus, $\mathcal{M}^{\Theta}, \sigma \models U\psi$.

The case $\varphi = \neg U\psi$. Assume that $\sigma\neg U\psi$ occurs on Θ. Since σ is an urfather and Θ is saturated, there is a prefix τ, such that $\tau\neg\psi$ occurs on Θ. Then, by (i) of Lemma 35 it follows that $u_{\Theta}(\tau)\neg\psi$ occurs on Θ, as well. But then, it follows from the induction hypothesis that $\mathcal{M}^{\Theta}, u_{\Theta}(\tau) \models \neg\psi$, which further implies that $\mathcal{M}, \sigma \models \neg U\psi$. □

From this truth lemma completeness easily follows:

Theorem 4 (Completeness for SSNL)**.** *If φ is valid in* SSNL*, then there is a tableau proof of φ.*

Proof. The proof goes by contraposition. Assume there is no tableau proof of φ. This means that there is tableau starting with $\sigma\neg\varphi$ that has an open saturated branch Θ. From this branch we can build the canonical model $\mathcal{M}^{\Theta}$. Finally, it follows from Lemma 36 that $\mathcal{M}^{\Theta}, \sigma \models \neg\varphi$, which further implies that φ cannot be valid. □