# Knowledge, Friendship and Social Announcements

JEREMY SELIGMAN, FENRONG LIU AND PATRICK GIRARD

ABSTRACT. We present a formal language for reasoning about the changing patterns of knowledge and friendship in social networks. A social announcement consists of an agent (the sender) transmitting some information (the message) to one or more other agents (the receivers) within a network and each of these three components can be described in different ways, from different perspectives. We discuss a number of conceptual issues that arise in such social communication and illustrate our ideas with a number of examples about cold-war spy networks and office gossip.

Love comes from blindness; friendship from knowledge.
–Comte De Bussy-Rabutin

## 1 Introduction

We will be interested in analysing scenarios in which there is a significant interaction between knowledge and social relationships. For example:

*Berlin 1978. A spy network has recently been uncovered by the Stasi, who are rounding up the spies and their associates. Bella is friends with Charlie and Erik, neither of whom are friends with each other. Unknown to the others is that Erik is a spy. The others are not spies, and Erik knows that because all spies know who else is a spy (we suppose). Bella knows that Charlie is not a spy, but Charlie does not know about her. After the network is exposed all the spies and their friends will be interrogated by the police. But just before this happens a message is relayed to all agents revealing whether or not they are in danger, that is, whether they are a spy (which they would know in any case) or a friend of a spy. Who now knows that Erik is a spy?*

To answer questions of this kind in such a scenario we need to reason about knowledge on the basis of social relationships. The purpose of this paper is to discuss a number of conceptual issues that arise when considering communication between agents in such networks, both from one agent to another, and broadcasts to socially-defined groups of agents, such as the group of my friends. We aim to provide a precise language for exploring 'logic in the community' [13] and reasoning in social networks in general. The framework we are going to propose is a combination of epistemic logic, dynamic logic and hybrid-like logic. In what follows, we will first review what has been developed in those fields and then position our contribution.

Epistemic logic (EL), with Hinttika's pioneering work [5] in philosophy and further developments in computer science and AI such as [7], is used to reason about knowledge. When more agents are involved, to talk about other agents' knowledge, EL was extended to multi-agent epistemic logic (multi-EL), with little change in the logic. A more significant extension was to include operators for common knowledge, in [3]. This produced a far richer agenda of topics and techniques for epistemic logic, in which such propositions as the following can be formalised:

- If I don't know that I'm in danger then I know that I don't know it. (EL)
- Bella knows that Erik knows that Charlie doesn't know that Erik is in danger. (multi-EL)
- It is not common knowledge that Erik is in danger. ($\mathsf{EL}^{\mathsf{C}}$)

Since the 1980s, when the information-driven dynamics of knowledge and belief came to the fore, interest shifted to how knowledge and beliefs change in response to new information. The AGM-paradigm [1] proposed rational postulates governing belief revisions. A slightly different framework, PAL, the logic of public announcement [9], analysed how an agent's information is updated in response to concrete action-like announcements. Typically public announcement results in common knowledge of the announced message, because each agent knows not only the content of the message, but that every other agent has received it. In dynamic epistemic logic, DEL, following [2], private announcements, in which a message is received by a limited set of agents are also considered. This extended the analysis of knowledge and communication to include such propositions as:

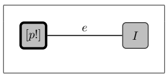
- Were it to be publicly announced that Erik is a spy, Charlie and Bella would know that. (PAL)
- If Bella does not know that she is in danger, and Charlie were told this in private, Bella would still not know it. (DEL)

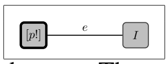
For uniformity of presentation, we will summarise the work in this area using the following partially specified formal language, based on a set Prop of propositional variables, a set $A$ of agents, and a set $\mathcal{D}$ of dynamic operators:

$$\varphi ::= \rho \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_\alpha\varphi \mid C_G\varphi \mid \Delta\varphi \quad (\rho \in \mathsf{Prop},\ \alpha \in A,\ G \subseteq A,\ \Delta \in \mathcal{D})$$

Here, $K_a\varphi$ means that agent $a$ knows that $\varphi$, $C_G\varphi$ means that it is common knowledge among the agents in the group $G$ that $\varphi$, and $\Delta\varphi$ (for $\Delta \in \mathcal{D}$) means that were action/event $\Delta$ to occur, $\varphi$ would be true. Details of the class $\mathcal{D}$ vary and some approaches are based on belief rather than knowledge, but the above will suffice for illustrative purposes.

With such a language, we can represent epistemic propositions such as the following:

| | |
|---|---|
| $K_e(p \to q)$ | Erik ($e$) knows that if ($p$) the network has been exposed then ($q$) there is a mole. |
| $K_b\neg K_c p$ | Bella ($b$) knows that Charlie ($c$) does not know the network has been exposed. |
| $C_{bc}p$ | It is common knowledge between Bella and Charlie that the network has been exposed. |
| $[p!]K_e q$ | If it were publicly announced that the network has been exposed, Erik would know there is a mole. |
| $(\neg K_e q \wedge$   $\neg K_e q)$ | Erik doesn't know that there is a mole, and were it to be announced privately to the others that the network has been exposed, he would still not know. |

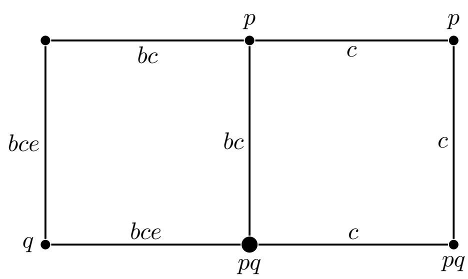
The last example uses the dynamic operator 

which represents the action of announcing $p$ to all agents other than $e$. The announcement of $p$ is represented by the node marked $[p!]$, which is highlighted as the action actually performed. But there is another possible action $I$, the 'identity' action in which nothing changes. Agent $e$'s ignorance about whether or not the announcement has taken place is represented by the line marked $e$. This is what makes the announcement, in some sense, private. Operators like these are called 'event models' or 'action models' in the literature, because of their Kripke-model-like appearance, but they are really part of the syntax of the language, not its semantics.

For semantics, a model for the language is a standard Kripke model with a binary accessibility relation for each modal operator: $M = \langle W, k, V\rangle$, where $W$ is a set of epistemic states and for each propositional variable $\rho \in \mathsf{Prop}$, the function $V$ assigns a set $V(\rho) \subseteq W$ of states in which $\rho$ holds. Then, for each agent $a \in A$, there is an equivalence relation $k_a$ on $W$, representing that agent's ignorance about which state she is in. If $k_a(u, v)$ then agent $a$ cannot determine of $u$ or $v$ is the actual state: for her they are 'epistemically indistinguishable'. For a group $G$ of agents, we define $k_G$ to be the transitive closure of $\bigcup_{a \in G} k_a$, so that two states $u$ and $v$ are related by $k_G$ if there is some path $u_0, \ldots, u_n$ connecting $u = u_0$ to $v = u_n$, such that for each link in the path, say from $u_i$ to $u_{j+1}$, there is at least one agent in the group who cannot distinguish between $u_i$ and $u_{j+1}$. Each dynamic operator $\Delta \in \mathcal{D}$ is associated with a transformation, mapping each model $M$ and state $w$ to a new model $\Delta M$ and state $\Delta w$, representing the result of performing some action. The details depend, of course, on the operator itself. Formulas are evaluated as follows:

| | | |
|---|---|---|
| $M, w \models \rho$ | iff | $w \in V(\rho)$ |
| $M, w \models \neg\varphi$ | iff | $M, w \not\models \varphi$ |
| $M, w \models (\varphi \wedge \psi)$ | iff | $M, w \models \varphi$ and $M, w \models \psi$ |
| $M, w \models K_a\varphi$ | iff | $M, v \models \varphi$ for all $v \in W$ such that $k_a(w, v)$ |
| $M, w \models C_G\varphi$ | iff | $M, v \models \varphi$ for all $v \in W$ such that $k_G(w, v)$ |
| $M, w \models \Delta\varphi$ | iff | $\Delta M, \Delta w \models \varphi$ |

Consider the model $M$ depicted below, with six epistemic states, among them the actual state $u$ is shown as larger than the others. Two propositions $p$ and $q$ are being considered and their truth value varies from state to state; states in which $p$ holds are marked with a '$p$'. The relations $k_a$ are represented as paths in the model, labelled with the name of the agent $a$. (There is a path from any point to itself, and if two or more lines marked '$a$' are joined, they form a path.) The relation $k_G$ for a group $G$ is therefore given by paths made up of lines that are labelled by at least one agent in $G$.

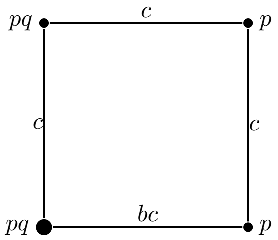



$$M, u \models K_e(p \to q)$$

$$M, u \models K_b \neg K_c p$$

$$M, u \not\models C_{\{bc\}} p$$

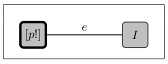
The classic example of a dynamic operator is that of publicly announcing some proposition. The public announcement $[p!]$ of $p$ transforms the model $M$ into the model $[p!]M$ in which all not-$p$ worlds are eliminated, as shown below. The actual state $[p!]u$ is just $u$ itself. We can then evaluate formulas $K_e q$ and $C_{\{bce\}} p$ in the new model and they are both true.



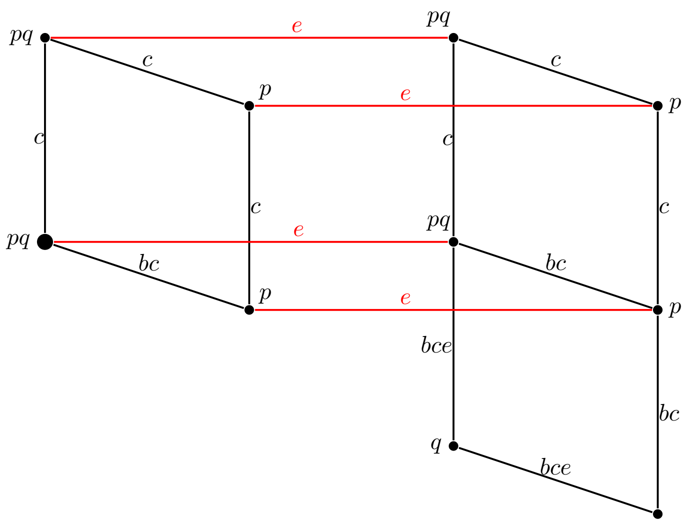

$M, u \models [p!]K_e q$ because $[p!]M, u \models K_e q$

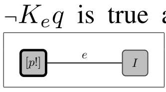
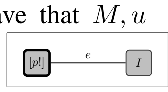
$M, u \models [p!]C_{\{bce\}} p$ because $[p!]M, u \models C_{\{bce\}} p$

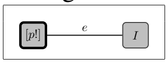
For private announcement, things are trickier. We review the main idea by considering the evaluation of our example formula:

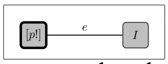
$(\neg K_e q \wedge$   $\neg K_e q)$

The dynamic operator acts on our model $M$ by combining the two models $[p!]M$ and $IM$, which is just $M$ itself, by adding links representing $e$'s ignorance about which model he is in. The actual state of the new model is the pair $\langle u, d\rangle$ where $u$ is the actual state of $M$ and $d$ is the designated node of the operator, representing the action actually performed.




In this new model, we can see that $q$ is false at the right bottom state, so $\neg K_e q$ is true at the actual state. Namely, we have that $M, u \models (\neg K_e q \wedge$   $\neg K_e q)$ because $M, u \models \neg K_e q$ and   $M, \langle u, d\rangle \models$

$\neg K_e q$. After an announcement to the other agents that the network is compromised, Erik would still not know it. Similarly, readers are invited to check the truth value of the following two formulas concerning common knowledge in the above model, too. It is not hard to see that in $M$ at $u$, the formula 

$C_{bc}p$ is true, whereas 

$C_{be}p$ is false, which is to say that were it to be announced to all agents other than Erik that the network is compromised, then this would be common knowledge among Bella and Charlie, but not among Bella and Erik.

The main concern of this paper is to extend this analysis to reasoning about knowledge and communication defined implicitly in terms of social relationships, most simply the binary relation of 'friendship', which we interpret in a minimal way as any symmetric irreflexive relation between agents on the basis of which exchange of information can occur. Some examples of propositions of this kind are the following:

- I know that all my friends are in danger but not all my friends know they are.
- Were Bella to tell all her friends whether or not they are in danger, she would know that Charlie knows he is not in danger.
- Were Erik to tell all his friends that he is a spy, Charlie would not know whether he is in danger.
- Were I to announce to all my friends that they are my friends, they would know this.

We will base our analysis on agent-indexical propositions, such as the first and last of the above, using names to refer to specific agents, and treating expression such as 'all my friends' as modal operators. The analysis of communication in such a setting involves careful attention to the perspective from which the communication is described. All three components of the communication (the sender, the message, and the receivers) can be specified in a variety of ways that need to be distinguished. In particular, the receivers of a message may be listed explicitly, or described as 'Bella's friends', or even as 'my friends', so that a sender may not know exactly who receives his message. Likewise, the content of the message may be about the sender or the receiver. For example, Charlie may tell Bella 'you are in danger' (about the receiver, Bella) or 'I am not a spy' (about the sender, Charlie). He may broadcast to all 'my friends are in danger', which if Bella is a friend, and does this, will enable her to infer that she is in danger, or send a message only to her friends that they are in danger. All such possibilities, together with their epistemic consequences, will be studied in subsequent sections.

The technical framework for this work is general dynamic dynamic logic (GDDL) [4], which provides a method for extending modal logics with dynamic operators for reasoning about a wide range of model-transformations, starting with those definable in propositional dynamic logic (PDL) and extended to allow for the more subtle operators involved in, for example, private communication, as represented in dynamic epistemic logic (DEL) and related systems. We provide a hands-on introduction to GDDL, introducing elements of the formalism as we go, showing how GDDL can be employed in a two-dimensional setting, but leave the reader to consult [4] for further details.

## 2 A language of social knowing

To represent the logical structure of propositions and reasoning about knowledge in a social context, specifically those involving friendship, we extend epistemic logic to EFL, an *epistemic logic of friendship*. The language is based on atoms of two types: propositional variables $\rho \in \mathsf{Prop}$ representing indexical propositions such as 'I am in danger', and (a finite set of) agent nominals $n \in \mathsf{ANom}$ which stand for indexical propositions asserting identification: 'I am $n$'. The language is then inductively defined as:

$$\varphi \quad ::= \quad \rho \mid n \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K\varphi \mid F\varphi \mid A\varphi \mid \; {\downarrow} n \, \varphi$$

We read $K$ as 'I know that', $F$ as 'all my friends', $A$ as 'every agent', ${\downarrow} n \, \varphi[n]$ as '$\varphi$ (me) holds from my perspective'.

As usual in modal logic we can define the duals of the operators, which we write inside angle brackets: $\langle K \rangle = \neg K \neg$ 'it is epistemically possible for me that', $\langle F \rangle = \neg F \neg$ 'I have a friend who', and $\langle A \rangle = \neg A \neg$ 'there is someone who'. We also use abbreviations for the hybrid-logic-like operators $@_n\varphi = A(n \rightarrow \varphi)$ (equivalently, $\langle A \rangle(n \wedge \varphi)$).[1] So, for example, if $\underline{n}$ is Charlie then the operator $@_n$ simply shifts the indexical subject to Charlie, so that $@_n d$ means 'Charlie is in danger'.

The English glosses are not so exact and require some manipulation to get proper translations because of the way pronouns work in English. For example, if $d$ represents 'I am in danger' then $\langle F \rangle K d$ means 'I have a friend who knows that he is in danger' rather than 'I have a friend who I know that I am in danger' which

[1] Although reminiscent of hybrid logic, the 'agent nominals' $n$, binder ${\downarrow} n$ and now the operator $@_n$ are not exactly the same as their hybrid-logic namesakes, but are rather some sort of two-dimensional cousins. A true nominal, for example, is a proposition that is logically compelled to be satisfied by exactly one evaluation index, which in the case of our models, would have to be the pair $\langle w, a \rangle$.

| | |
|---|---|
| $\neg K@_e s$ | I don't know that Erik ($e$) is a spy ($s$) |
| $Fd$ | All my friends are in danger |
| $KFd$ | I know that all my friends are in danger |
| $@_e KFd$ | Erik knows that all his friends are in danger |
| $FKd$ | All my friends know they are in danger |
| $\langle F\rangle d$ | Some of my friends are in danger |
| $\langle F\rangle c$ | Charlie ($c$) is my friend |
| $\langle F\rangle K@_e d$ | Some of my friends know that Erik is in danger |
| $\downarrow n\ \langle F\rangle K@_n d$ | I have a friend who knows that I am in danger |

Figure 1. Some statements of EFL

is not even grammatically correct! The hybrid feature of the language enables us to express indexical propositions. Finally, $\downarrow n\ FK\langle F\rangle n$ says 'all my friends know they are friends with me'. This provides a way of referring to 'me' inside the scope of other operators, by shifting the referent of $n$ to the current agent. To illustrate, we give more examples in Figure 1.

Models for this language are two-dimensional Kripke models of the form $M = \langle W, A, k, f, V\rangle$, where $W$ is a set (of epistemic states), $A$ is a set (of agents), and

1. $k$ is a family of equivalence relations $k_a$ for each agent $a \in A$, representing the ignorance of $a$ in distinguishing epistemic possibilities (as for standard S5 epistemic logic)

2. $f$ is a family of symmetric and irreflexive relations $f_w$ for each $w \in W$, representing the friendship relation in state $w$.

3. $g$ is a function mapping each agent nominal $n \in \mathsf{ANom}$ to the agent $g(n) \in A$ named by $n$. We abbreviate $g(n)$ to $\underline{n}$ when the model is clear from the context.

4. $V$ is a valuation function mapping propositional variables Prop to subsets of $W \times A$, with $(w, a) \in V(p)$ representing that the indexical proposition $p$ holds of agent $a$ in state $w$.

In order to interpret the $\downarrow$operator, we introduce a slightly different mapping: For $m \in \mathsf{ANom}$,

$$g[^n_a](m) = \begin{cases} a & \text{if } m = n \\ g(m) & \text{otherwise} \end{cases}$$

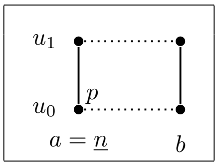



Figure 2. A simple EFL model

Models $\langle W, A, k, f, g[^n_a], V\rangle$ based on this mapping is denoted as $M[^n_a]$, which is the result of changing $M$ so that $n$ now names $a$.

For example, Figure 2 illustrates a simple model for a language in which there is only one propositional variable $p$ and one agent name $n$. The set of states is $W = \{u_0, u_1\}$ and the set of agents is $A = \{a, b\}$, with $g(n) = a$, $n$ naming agent $a$. Both agents are ignorant about which state they are in, so $k_a = k_b$ is the universal relation. These are indicated by the two columns of the diagram. The left column displays the $k_a$ relation with a thick line; the right column displays the $k_b$ relation, similarly. The lines are non-directional because the relations are assumed to be symmetric. In more complex diagrams, we will assume that the relations depicted are the reflexive, transitive closures of what is shown explicitly. The rows of the diagram show the relations $f_{u_0}$ (first row) and $f_{u_1}$ (second row) with dotted lines. This represents the two agents being friends in both states of $W$. Again these are non-directional because we assume symmetry. But for these lines we do *not* take the reflexive, transitive closure, since we assume that $f_w$ is irreflexive and may or may not be transitive. Finally, that $p$ holds only of agent $a$ in state $u_0$, i.e., that $V(p) = \{(u_0, a)\}$ is shown by labelling the lower left node of the diagram with $p$.

Models are used to interpret $\mathcal{L}$ in a double-indexical way, as follows:

| | | |
|---|---|---|
| $M, w, a \models \rho$ | iff | $(w, a) \in V(\rho)$, for $\rho \in \mathsf{Prop}$ |
| $M, w, a \models n$ | iff | $g(n) = a$, for $n \in \mathsf{ANom}$ |
| $M, w, a \models \neg\varphi$ | iff | $M, w, a \not\models \varphi$ |
| $M, w, a \models (\varphi \wedge \psi)$ | iff | $M, w, a \models \varphi$ and $M, w, a \models \psi$ |
| $M, w, a \models K\varphi$ | iff | $M, v, a \models \varphi$ for every $v \in W$ such that $k_a(w, v)$ |
| $M, w, a \models F\varphi$ | iff | $M, w, b \models \varphi$ for every $b \in A$ such that $f_w(a, b)$ |
| $M, w, a \models A\varphi$ | iff | $M, w, b \models \varphi$ for every $b \in A$ |
| $M, w, a \models \downarrow n\, \varphi$ | iff | $M[^n_a], w, a \models \varphi$. |

We say that $M$ is a *named agent* model, if every agent in $M$ has a name, i.e., for each $a \in A$, there is an $n \in \mathsf{ANom}$ such that $g(n) = a$. The model depicted in Figure 2 is *not* a named agent model because agent $b$ has no name. Our discussion in this context will not be restricted to named agent models.

### 2.1 Transforming models with PDL

We will define a class of operators $\mathcal{D}$ and corresponding actions on models such that for each $\Delta \in \mathcal{D}$ and each $M$ model for $\mathcal{L}$, there is an $\mathcal{L}$ model $\Delta M$, and for each state $w$ of $M$, a state $\Delta w$ of $\Delta M$. We then extend $\mathcal{L}$ to a language $\mathcal{L}(\mathcal{D})$ of *dynamic epistemic friendship logic* (DEFL) by adding the elements of $\mathcal{D}$ as propositional operators and defining

$$M, w, a \models \Delta\varphi \qquad \text{iff} \quad \Delta M, \Delta w, a \models \varphi$$

To define $\mathcal{D}$, we use the language of propositional dynamic logic (PDL) with basic programs $K$, $F$ and $A$, given by

$$\begin{array}{ll} \mathcal{T} & \pi ::= K \mid F \mid A \mid \varphi? \mid (\pi;\pi) \mid (\pi \cup \pi) \mid \pi^* \\ \mathcal{F} & \varphi ::= \rho \mid n \mid \neg\varphi \mid (\varphi \vee \varphi) \mid \langle\pi\rangle\varphi \end{array}$$

for $\rho \in$ Prop and $n \in$ ANom. The denotation of program terms $\pi \in \mathcal{T}$ and formulas $\varphi \in \mathcal{F}$ in a model $M$ are defined in the manner shown in Table 1. Note

$$\begin{array}{lll}
[\![\rho]\!]^M & = & V(\rho), \text{ for } \rho \in \mathsf{Prop} \\
[\![n]\!]^M & = & W \times \{g(n)\}, \text{ for } n \in \mathsf{ANom} \\
[\![(\varphi \wedge \psi)]\!]^M & = & [\![\varphi]\!]^M \cap [\![\psi]\!]^M \\
[\![\neg\varphi]\!]^M & = & W \setminus [\![\varphi]\!]^M \\
[\![\langle\pi\rangle\varphi]\!]^M & = & \{w \in W \mid w[\![\pi]\!]^M v \text{ and } v \in [\![\varphi]\!]^M \text{ for some } v \in W\} \\
[\![K]\!]^M & = & \{\langle(w,a),(v,a)\rangle \mid k_a(w,v)\} \\
[\![F]\!]^M & = & \{\langle(w,a),(w,b)\rangle \mid f_w(a,b)\} \\
[\![A]\!]^M & = & \{\langle(w,a),(w,b)\rangle \mid a,b \in A,\ w \in W\} \\
[\![\varphi?]\!]^M & = & \{\langle w,w\rangle \mid w \in [\![\varphi]\!]^M\} \\
[\![\pi_1;\pi_2]\!]^M & = & \{\langle w,v\rangle \mid w[\![\pi_1]\!]^M s \text{ and } s[\![\pi_2]\!]^M v \text{ for some } s \in W\} \\
[\![\pi_1 \cup \pi_2]\!]^M & = & [\![\pi_1]\!]^M \cup [\![\pi_2]\!]^M \\
[\![\pi^*]\!]^M & = & \{\langle w,v\rangle | w = v \text{ or } w_i[\![\pi]\!]^M w_{i+1} \text{ for some } n \geq 0, \\
& & w_0,\ldots,w_n \in W, w_0 = w \text{ and } w_n = v\}
\end{array}$$

Table 1. Semantics of PDL terms and formulas

in particular, the clauses for $K$, $F$ and $A$, in which these program terms refer to the accessibility relations of the corresponding operators of EFL, when interpreted two-dimensionally. Complex program terms are built up in the usual way: $(\pi_1;\pi_2)$ for the relational composition of $\pi_1$ and $\pi_2$, $(\pi_1 \cup \pi_2)$ for their union (or choice), $\varphi?$ for the 'test' consisting of a link from $(w,a)$ to itself iff $M, w, a \models \varphi$, and $\pi^*$ for the reflexive, transitive closure of $\pi$, which is understood as a form of iteration.

Note also that we have abused notation so that formulas $\varphi$ of EFL, written with existential operators $\langle K\rangle$, $\langle F\rangle$ and $\langle A\rangle$, are also programs formulas (in $F$). This is justified by the obvious semantic equivalence:

$$M, w, a \models \varphi \text{ iff } (w, a) \in [\![\varphi]\!]^M$$

Now the class of dynamic operators will be defined using the theory of General Dynamic Dynamic Logic (GDDL) given in [4], which provides an extension to any language of PDL. The simplest of these operators are called PDL-*transformations*. These consist of assignment statements which transform models by redefining the basic programs. For example, the operator $[K := \pi]$ acts on model $M$ to produce a new model $[K := \pi]M$ such that

$$[\![K]\!]^{[K:=\pi]M} = [\![\pi]\!]^M$$

On states, there is no change: $[K := \pi]w = w$, so the resulting DEFL operator has the following semantics:

$$M, w, a \models [K{:=}\pi]\varphi \quad \text{iff} \quad [K{:=}\pi]M, w, a \models \varphi$$

We must be a little careful in the choice of $\pi$ so as to ensure that the resulting model $[K := \pi]M$ is still a model for EFL. For example, consider the program term $n?; K$. In $M$, this relates $(u_0, a)$ to $(u_1, b)$ in case $(u_0, a) \in [\![n]\!]^M$ and $(u_0, a)[\![K]\!]^M(u_1, b)$, which only holds when $g(n) = a$, $a = b$, and $k_a(u_0, u_1)$. Then $[K := n?; K]M$ is the structure $\langle W, A, k', f, V\rangle$ in which $k'_a = k_a$ and $k'_b = \emptyset$, for $b \neq a$. This is *not* a model for EFL. To make it into a model for EFL, we need to make each $k_a$ reflexive. This can be done with the program term $\top?$, since $[\![\top?]\!]^M$ is the identity relation. Thus taking $\pi$ to be $(n?; K) \cup \top?$ we get the model $[K := (n?; K) \cup \top?]M$ which is the structure $\langle W, A, k'', f, V\rangle$ in which $k''_a = k_a$ and $k''_b$ is the identity relation for all $b \neq a$. The application of $[K := (a?; K) \cup \top?]$ to a particular model is illustrated in Figure 3. Here, $M$ is a named agent model, so we allow ourselves the abuse of notation involved in writing $a$ for the name of $a$. In this model there are two friends, $a$ and $b$, who are both ignorant about whether they are in state $u_0$ or $u_1$. $p$ holds only of agent $a$ in state $u_0$, so in particular, $M, u_0, b \models (K\neg p \wedge \neg K\langle F\rangle p)$, which means that agent $b$ knows that she is not $p$ but does not know whether she has a friend who is $p$. After the action $[K := (n?; K) \cup \top?]$ we get the model shown on the right, in which $k_a$ is as before but now $k_b$ is the identity relation. In the transformed model, agent $b$ now knows that she has a friend who is $p$. Thus we get the dynamic fact:

$$M, u_0, b \models [K := (n?; K) \cup \top?]K\langle F\rangle p$$

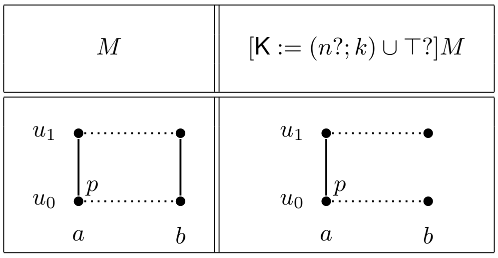



Figure 3. A simple PDL-transformation.

In effect, the PDL-transformation, $[K := (n?; K) \cup \top?]$ is the action of revealing everything to every agent other than $\underline{n}$. We will consider more subtle forms of epistemic change in subsequent sections. Now it is time for an analysis of the spy network example in the Introduction. We repeat the story here.

> *Berlin 1978. A spy network has recently been uncovered by the Stasi, who are rounding up the spies and their associates. Bella (b) is friends with Charlie (c) and Erik (e), neither of whom are friends with each other. Unknown to the others is that Erik is a spy (s). The others are not spies, and Erik knows that because all spies know who else is a spy (we suppose). Bella knows that Charlie is not a spy, but Charlie does not know about her. After the network is exposed, all the spies and their friends will be interrogated by the police. But just before this happens a message is relayed to all agents revealing whether or not they are in danger, that is, whether they are a spy (which they would know in any case) or a friend of a spy. Who now knows that Erik is a spy?*

A model $M$ of the initial situation is depicted in Figure 4, with $u_0$ representing the actual state. In EFL we can state pertinent facts such as $@_b(K\neg s \wedge \neg K\langle F\rangle s)$ 'Bella knows that she is not a spy but doesn't know if a friend of hers is a spy'. We will write $d$ 'I am in danger' as an abbreviation for $(s \vee \langle F\rangle s)$ 'either I'm a spy or I have a spy as a friend', and, for convenience, we have labelled those state-agent pairs at which $d$ holds. Thus we can read that $@_b(d \wedge \neg Kd)$ 'Bella is in danger but doesn't know it', whereas $@_b K @_c \neg d$ 'Bella knows that Charlie is not in danger'.

Now consider the PDL-term $\mathsf{cut}_K(\varphi)$ defined by

$$(\varphi?; K; \varphi?) \cup (\neg\varphi?; K; \neg\varphi?)$$

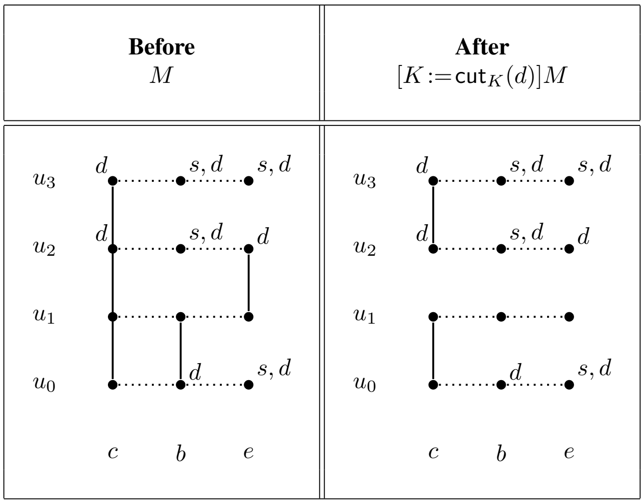



Figure 4. Spy Network

This relates $\langle w, a\rangle$ to $\langle v, b\rangle$ iff $a = b$, $k_a(w, v)$, and either $\varphi$ is true of $a$ in both states $w$ and $v$ or false of $a$ in both states. Thus the operator $[K := \mathsf{cut}_K(\varphi)]$ produces a new model $[K := \mathsf{cut}_K(\varphi)]M$ from $M$ by removing the $k_a$ links between states with conflicting values for $\varphi$ (about $a$). Effectively, this 'reveals' to each agent whether or not $\varphi$ holds (for them).[2]

In our example, the situation after the revelation of $d$ 'you are in danger' is given by the model $[K := \mathsf{cut}_K(d)]M$, shown in the right part of Figure 4. Notice that the $k_c$ link between $u_1$ and $u_2$ are cut because $M, u_1, c \not\models d$ but $M, u_2, c \models d$; Charlie finds out that he is not in danger. Similarly, the $k_b$ link between $u_0$ and $u_1$ is cut because Bella finds out that she *is* in danger ($@_b Kd$). Finally, the $k_e$ link between $u_1$ and $u_2$ is cut because everyone now knows that Erik knows whether he is in danger (although only Bella knows which). Reasoning about such changes can be represented in the language of DEFL such as the valid schema

$$[K := \mathsf{cut}_K(\varphi)]A(K\varphi \vee K\neg\varphi)$$

which states (for non-epistemic facts $\varphi$ such as $d = \langle F\rangle s$) that after $\varphi$ is revealed, everyone knows whether $\varphi$ or not.

After the same update we can also ask who then knows that Eric is a spy (and did not know before)? Formally put in DEFL,

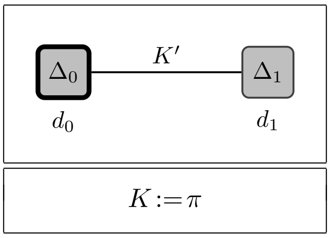
[2]This operator was first introduced in [15].

For which $n$, $@_n(\neg K@_e s \wedge [K := \mathsf{cut}_K(d)]K@_e s)$?

Again reading from the right part of the Figure 4, we see that it is Bella who knows that Eric is a spy, as Charlie is still uncertain.

## 2.2 GDDL operators

More complicated operators can be constructed from finite relational structures whose elements are each associated with a PDL transformation, and whose combined effect on the model is calculated by 'integrating' them according to a further transformation. A GDDL operator $\Delta$ is something that looks like this:

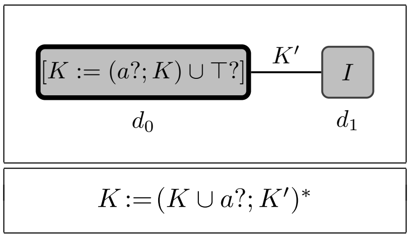



This represents an action $d_0$ (highlighted as the action that is actually performed) whose effect on the model is given by the PDL-transformation $\Delta_0$. There is also an action $d_1$ with associated PDL-transformation $\Delta_1$, and the relationship between $d_0$ and $d_1$ is marked as $K'$.[3] The effect of the operator on an EFL model $M$ with domain $W$ is computed by forming a product model $M'$ (in the manner of [2]) whose domain is $W \times \{d_0, d_1\}$, in which the elements $(w, d_i)$ represent the state resulting from action $d_i$ when the initial state is $w$. The model $M'$ consists of copies of two models $[\Delta_0]M$ with domain $W \times \{d_0\}$ and $[\Delta_1]M$ with domain $W \times \{d_1\}$, and a duplication of the model occurring in $\Delta$ itself, with, in this case, $(w, d_0)[\![K']\!]^{M'}(w, d_1)$ for each $w, v \in W$. Finally, the model $[\Delta]M$ is computed by applying the 'integrating' transformation $[K := \pi]$ to $M'$. This uses a PDL program term $\pi$ to compute the new value for $K$ from a combination of relations in the copied models $[\Delta_0]M$ and $[\Delta_1]M$ and the new relation $K'$ from $\Delta$ itself.[4]

This somewhat complex operation is best explained by looking at a simple example. Consider the case in which $\Delta_0$ is the PDLtransformation $[K := (a?; K) \cup \top?]$ considered earlier, and $\Delta_1$ is the identity transformation, $I$. We will also take $\pi$ to be $(K \cup a?; K')^*$.

[3] In the general case, as explained in [4], there may be many actions and many new relation symbols; also, propositional variables.

[4] Again, the general case is more flexible, allowing any of the basic expressions $K$, $F$, agent nominal and propositional variable to be reinterpreted at the integrating stage.




The action of this GDDL operator on the model $M$ considered earlier, is shown in Figure 5. It represents a situation in which an action $d_0$ gives complete information to all agents other than $a$. The occurrence of $d_0$ is known to all agents other than $a$, who stays completely in the dark. Not only is $k_a$ unchanged in both $[\Delta_0]M$ (the top half of the diagram) and $[\Delta_1]M$ (the bottom half), but $a$ is also ignorant about which of these two submodels she is in, as represented by the vertical lines in connecting the two halves of the $a$ column: $(w, d_0)k_a(w, d_1)$ for all $w \in W$. Once

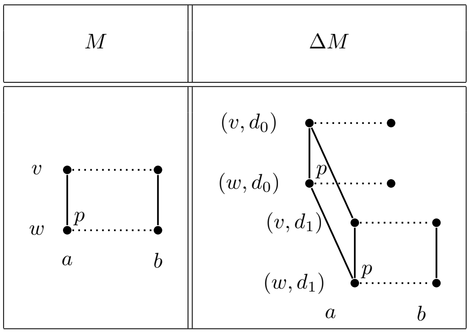



Figure 5. A simple GDDL operator in action.

again, we must check that the resulting model is an EFL model. In this case, it is. The $k_a$ and $k_b$ relations are transitive thanks to the application of the $*$ operator in the integrating transformation $[K := (K \cup a?; K')^*]$.

We'll say that a GDDL-transformation $\Delta$ is a *general* EFL *dynamic operator* if it is in the language of PDL terms defined above, possibly augmented with internal relations such as $K'$ and also preserves the property of being a EFL-model: whenever $M$ is an EFL-model, so is $\Delta M$. Further work is needed to characterise this class syntactically.

## 3 Social announcements

It is time to consider direct communications, or 'announcements', within a social network. As we stated before, an announcement in a social network consists of three components (i) the sender (ii) the receiver(s) and (iii) the dynamic transmitting of some message.[5] In this section, we will study each component and try to capture the subtleties of social communication, in particular, when the indexical propositions are involved.

As a starting point, we ignore the sender and define a basic act of communication in which a message $\psi$ is sent (anonymously, we suppose) to a group of agents described by formula $\theta$ by

$$\mathsf{send}_\theta(\psi) = [K := (\theta?; \mathsf{cut}_K(\psi)) \cup (\neg\theta?; K)]$$

The action $\mathsf{send}_\theta(\psi)$ reveals the truth or falsity of $\psi$ (which may be different for different agents) to all agents satisfying $\theta$, and leaves the $k_a$ relation unchanged for agents $a$ not satisfying $\theta$.

To see how this works, consider $\mathsf{send}_{\langle F\rangle b}(d)$ in the case of our spy network. This is an anonymous announcement to the friends of Bella (but not to Bella herself) whether or not they are in danger. The effect of this action is shown in Figure 6. The formula $\theta$ describing the receivers of the message is $\langle F\rangle b$, which is satisfied by Charlie and Erik in the actual state $u_0$. Thus only the relations $k_c$ and $k_e$ are changed; $k_b$ remains the same. This is by no means our final analysis of communication. For one thing, actions of this sort are only 'semi-private', i.e., directed at particular individuals, but with others not involved in the communication still aware that it has occurred. Later, we will need to make the analysis more complex to cope with a great degree of privacy, in which only the sender and receivers are aware that the communication has occurred. For example, after the communication to Bella's friends, Bella knows something that she didn't know before: before she knew that Charlie was not in danger, now she knows that Charlie knows this:

$$M, u_0, b \models [\mathsf{send}_{\langle F\rangle b}(d)]K@_c K\neg d$$

Yet before we get to the issue of privacy, we will bring the sender into our model, and explore some subtle distinctions about the nature of the message itself.

[5] We are aware of the attempts by others in this respect. [10] analysed specific types of communication network (i.e., communications that take place between one agent and another, or between an agent and a group of agents) when considering the issue of how distributed knowledge can be established by a group of agents through communication. Communication graphs were adopted by [8] to study communication between agents. Agent $i$ directly receiving information from agent $j$ is represented by an edge from agent $i$ to agent $j$ in such graph. Neither approach considers groups of agents described in terms of social relations.

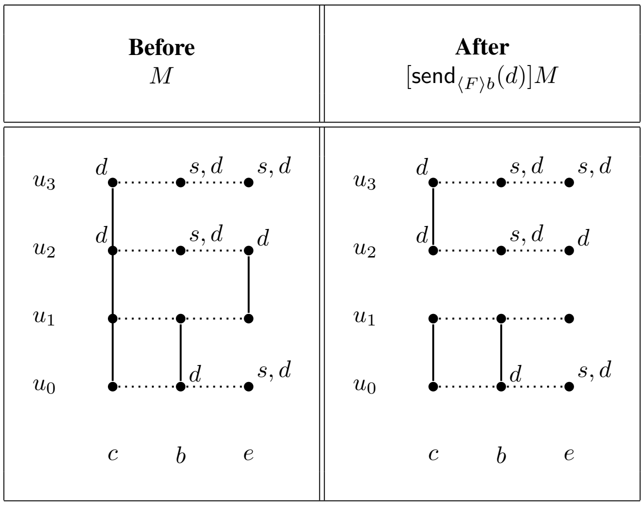



Figure 6. Restricting to Bella's friends

### 3.1 Announcements about the sender

The first case is that of a message sent by agent $n$ to *agents described by* $\theta$ with a message $\psi$, which is understood to be about the sender, e.g. 'I am in danger'. We define $[n \lhd \psi! : \theta]\varphi$, the statement that $\varphi$ would hold after such a communication, as

$$(@_n K\psi \to [\mathsf{send}_\theta(@_n\psi)]\varphi)$$

To make sense of this, we will look at a progression of simpler cases. First, with $\theta = \top$, the formula $[n \lhd \psi! : \top]\varphi$ means that $\varphi$ would hold were agent $n$ publicly to announce that $\psi$, noting that it simplifies to $(@_n K\psi \to [K := \mathsf{cut}_K(@_n\psi)]\varphi)$.

We make the rather strong assumption that the message is known by the sender.[6] Suppose, for example, that Erik, unable to keep his secret any longer, told everyone that he is a spy. After this, everyone would know that he is a spy (and Bella, his friend, would know that she is in danger). This follows from the validity of $[e \lhd s! : \top]AK@_e s$.[7] Note that $[b \lhd s! : \top]AK@_b s$ is also true (since it is valid!). This says that everyone would know that Bella is a spy were she to announce it. But the reason is quite different: Bella could not announce that she is a spy,

[6]The standard assumption of PAL that announcements are true is thus equivalent to supposing that they are made by God, or some other omniscient entity. [6] studied different types of agent (truth-teller, liar and bluffer) , how they make announcements, and are subsequently interpreted in communication.

[7]In fact, the information that Erik is a spy becomes common knowledge.

because she knows that she isn't.[8]

The second case is an announcement to *a particular agent*. In this case, $\theta$ is an agent nominal $m$ and the formula $[n \lhd \psi! : m]\varphi$ means that $\varphi$ would hold were agent $n$ to announce to $m$ that $\psi$. For example, Erik may be more cautious in his admission, telling only Bella, after which she, but not Charlie would know: $[e \lhd s! : b]@_b K @_e s$ and $(\neg(b \vee K@_e s) \to [e \lhd s! : b]\neg K @_e s)$ are both valid, and the latter says that an agent who is neither Bella nor (already) knows that Erik is a spy, still doesn't know this after he announces it to Bella.

In the most general case, $\theta$ is a description of a group of agents. For example, $[b \lhd \neg s! : \langle F \rangle b]\varphi$ states that $\varphi$ would hold after Bella tells her friends that she is not a spy. Again we have a useful validity: $[b \lhd \neg s! : \langle F \rangle b]@_b F K @_b \neg s$, which says that if Bella were to tell her friends that she is not a spy then they would all know that she isn't a spy.

### 3.2 Announcements about the receivers

Announcements that are indexical about the receiver such as 'you are in danger' (announced to Bella by Erik) or 'you are my friends' (announced by Bella to her friends) can be expressed with a slight change that captures the different preconditions for announcements of this kind. We define $[n : \psi! \rhd \theta]\varphi$, the statement that $\varphi$ holds after agent $n$ announces message $\psi$ (about $\theta$) to agents satisfying $\theta$ as

$$(@_n KA(\theta \to \psi) \to [\mathsf{send}_\theta(\psi)]\varphi)$$

Again, we first consider the simple case of public announcement, represented by $[n : \psi! \rhd \top]\varphi$, which can be seen to be equivalent to $(@_n KA\psi \to [K := \mathsf{cut}_K(\psi)]\varphi)$. Consider, for example, my announcing to everyone 'you are in danger'. The precondition that I know everyone is in danger is captured by the antecedent $KAd$, and after the announcement everyone knows that she is in danger, as is represented by the validity of $\downarrow n\, [n : d! \rhd \top]AKd$.

The case of agent-to-agent announcement displays a nice symmetry between the two kinds of indexical message. Agent $n$ announcing 'you are in danger' to agent $m$ is equivalent to announcing (again to $m$) that $m$ is in danger. More generally, the following equivalences are valid

$$[n : \psi! \rhd m]\varphi \;\leftrightarrow\; [n \lhd @_m \psi! : m]\varphi$$
$$[n \lhd \psi! : m]\varphi \;\leftrightarrow\; [n : @_n \psi! \rhd m]\varphi$$

This symmetry between announcements is more delicate when announcing to groups. Announcing 'you are in danger' to each of my friends is only the same as

[8]It would be enough for Bella merely not to know that she is a spy for the announcement to be impossible.

announcing to them ‘all my friends are in danger’ on the assumption that each friend knows only that she is my friend, and knows nothing about the others. Without this assumption,

$$[n\!:\psi!\rhd\langle F\rangle n]\varphi \;\leftrightarrow\; [n\lhd @_n F\psi!\!:\langle F\rangle n]\varphi$$

is not always valid.[9]

For announcement to friends, an interesting new phenomenon arises. Consider the case of my announcing ‘you are my friend’ to my friends. That $\varphi$ holds after such an announcement is represented by $[n\!:\langle F\rangle n!\rhd\langle F\rangle n]$. The message is the same as the description of the set of receivers, so when this is expanded, we find that the precondition for the announcement is $\downarrow\! n\; KA(\langle F\rangle n \to \langle F\rangle n)$, which is valid, so the announcement can always be made, by anyone. But nonetheless, it can be informative, as can be seen from the validity of $\downarrow n\; [n\!:\langle F\rangle n!\rhd\langle F\rangle n]FK\langle F\rangle n$, which says that after my making this announcement, my friends all know that they are my friends, something they may not have known before.

Finally, we note that any sender-indexical announcement to a group $\theta$ is equivalent to a receiver-indexical announcement to the same group $\theta$ in the case that there is at least one receiver ($A\neg\theta$ is false). The trick is that the statement $\psi$ about $n$ (the sender) is then equivalent to the statement $@_n\psi$ about any (every) receiver. More formally, the following is valid:[10]

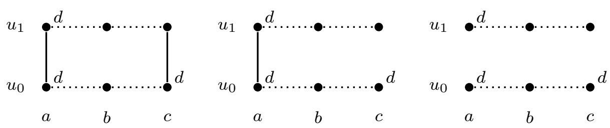
$$(\neg A\neg\theta \to [n\lhd\psi!\!:\theta]\varphi \;\leftrightarrow\; [n\!:@_n\psi!\rhd\theta]\varphi)$$

### 3.3 Private announcements

Communications of the form $[n\lhd\psi!\!:\theta]$ and $[n\!:\psi!\rhd\theta]$ are only semi-private. Their effect on the model ensures that every agent will know that the announcement has occurred, if the sender satisfies the precondition, so, for example,

$$\downarrow\! n\; [n\lhd d!\!:m]AK(@_nKd \to @_mK@_nd)$$

[9] For a simple counterexample, consider $\psi$ to be $d$ and the model $M$ (shown left).

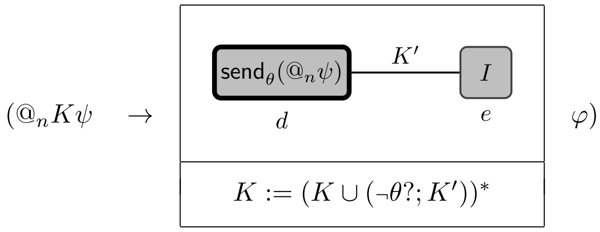


The precondition of $[b\!:d!\rhd\langle F\rangle b]$ is $@_bKA(\langle F\rangle b \to d)$, which is equivalent to the precondition $@_bKFd$ of $[b\lhd @_bFd!\!:\langle F\rangle b]$ which is satisfied in $M$, and the resulting two models are shown middle and right. Yet these are easily distinguished, by taking $\varphi$ to be $@_aK@_cd$.

[10] The key observation here is that the precondition for the sender-indexical announcement is $@_nK\psi$, which is equivalent to the precondition $@_nKU_A(\theta \to @_n\psi)$ when $U_A\neg\theta$ is false.

is valid: after I announce to $m$ that I am in danger, everyone will know that if I know I am in danger then $m$ also knows it. This is (typically) an unjustified violation of the privacy of the communication between me and $m$.

To make the action $\mathsf{send}_\theta(\psi)$ private, we embed it in a GDDL operator similar to the one given in our earlier example. Thus, for the sender-indexical[11] version, that $\varphi$ would hold after the private announcement of $\psi$ by $n$ to agents $\theta$ is be represented as




Call this formula $[\![n \triangleleft \psi! : \theta]\!]\varphi$. Inside the GDDL operator, the internal relation $K'$ represents ignorance about whether the communication $\mathsf{send}^n_\theta(\psi)$ has occurred or not, the latter possibility represented by the identity transformation, $I$. The integrating transformation $[K := (K \cup (\neg\theta?; K'))^*]$ restricts ignorance of the $K'$ kind to agents other than $\theta$ and factors this in to the new epistemic relation. The $*$ is needed to ensure that the result is an equivalence relation. We will see an example of this operator in action at the end of the next section.

## 4 Knowing your friends

So far, the friendship relation in our models has been relatively tame, remaining fixed across epistemic states. We have used it to determine which group of agents receive a message, and even to specify the content of a message, but we have not yet considered ignorance about who is friends with whom. This is where it gets really interesting. We will explore some of the possibilities with an everyday example of infidelity and gossip.

> *Peggy ($p$) knows that Roger ($r$) is cheating ($c$) on his wife, Mona ($m$). What's more, Roger knows that Peggy knows, because they met accidentally while he was with his mistress. Mona does not know about the affair, and both Peggy and Roger know this. The situation (for Roger) deteriorates when he discovers that Peggy is a terrible*

[11] The receiver-indexical version is obtained by changing the message and the precondition as in the simple semi-private case.

*gossip. She is bound to have told all her friends about his affair. What Roger does not know is whether Mona is a friend of Peggy (she is).*

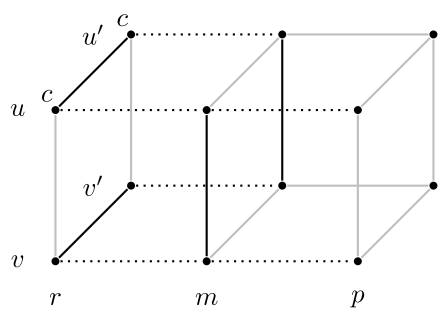



Figure 7. Roger's Quandry

We can represent the epistemic state of this network before Peggy's announcement with the model depicted in Figure 7, assuming that married couples are also friends. (The grey construction lines are only included to make the diagram easier to read; they have no epistemic or social significance.) Note that the friendship relations are now different in different states. At $u$ (the actual state) for Roger $r$, the statements listed in Table 2 are all true. As a result, we can compute that at $w$ in the original model for Roger $r$, the formula

$$\downarrow n\ [p \triangleleft @_n c! : \langle F \rangle p] @_m K @_n c$$

is true, i.e., "I don't know that Mona will know about my cheating after Peggy tells her friends about it." That some proposition $\varphi$ holds after the announcement 'Roger is cheating!' that Peggy makes to her friends is given by $[p \triangleleft @_r c! : \langle F \rangle p]\varphi$, which expands and simplifies to

$$(@_p K @_r c \rightarrow [K := (\langle F \rangle p?; \mathsf{cut}_K(@_r c)) \cup (\neg\langle F \rangle p?; K)]\varphi)$$

When evaluated at $u$, the presupposition that Peggy knows that Roger is cheating is satisfied, and so the formula $\varphi$ is evaluated in the transformed model shown in Figure 8. (Note the missing vertical line in the middle.)

This is all very well, but Roger needs a little more privacy.

*Before returning home to face Mona, Roger is uneasy. He would really like to know whether or not she knows about his affair. He*

| | |
|---|---|
| $c$ | I'm cheating |
| $\downarrow n\ K(@_p K @_n c \wedge @_m \neg K @_n c)$ | I know that Peggy (but not Mona) knows I am cheating. |
| $\downarrow n\ @_p K @_n K @_p K @_n c$ | Peggy knows I know she knows I am cheating |
| $\neg K @_m \langle F \rangle p \wedge \neg K @_m \neg \langle F \rangle p$ | I don't know whether Peggy and Mona are friends. |
| $\downarrow n\ @_p K @_n \neg K @_m \langle F \rangle p$ | Peggy knows I don't know whether she and Mona are friends. |

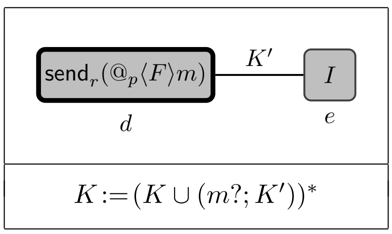
Table 2. Facts about Roger

*already knows that she knows if and only if she is friends with Peggy. So if Peggy told him that they are friends, he would be prepared for Mona's fury. But for his planned excuses to be convincing, Mona must not know that he knows she knows (about the affair). It is therefore very important that Peggy tells him in private.*

Now let us suppose that the ever-loquacious Peggy announces to Robert privately that Mona is her friend, represented as $[\![ p \triangleleft \langle F \rangle m! : r ]\!]$. Now, whether the crucial proposition $\varphi$

$$(@_r K @_m K @_r c \wedge \neg @_m K @_r K @_m K @_r c)$$

(that Roger knows Mona knows he has been cheating but Mona doesn't know that he knows) holds must be determined by evaluating it in the model obtained by transforming the one in Figure 8 using the following GDDL operator, call it $\Delta$:



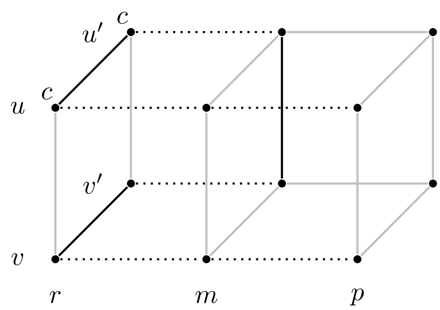



Figure 8. After Peggy's gossip

The result is shown in Figure 9.

The upper half of the diagram represents the result of action $d$, Peggy telling Roger that she is friends with Mona ($\mathsf{send}_r(@_p\langle F\rangle m)$), whereas the lower half represent the result of action $e$, nothing $(I)$; it is just a copy of the model in Figure 8. Mona is the only one of the three who doesn't know which action has taken place, and her ignorance is represented by the lines connected corresponding states in the upper and lower halves (in the $m$ column). We see that $K@_m K@_r c$ holds of $r$ in state $(u, d)$, so Roger can meet Mona prepared.[12]

We may wonder about the accuracy of the model in representing Roger and Mona as friends after Peggy's announcement. This will probably lead to changes to the social network, Roger and Mona may no longer be friends. We leave such issues for other occasions.

## 5 Conclusions

What has emerged from this study is an appreciation of the diversity of subtle logic distinctions when combining epistemic and social relations, especially when allowing indexical propositions, as are very common in the social setting. The patterns of inference we commonly use when reasoning about everyday social situations have been shown to be more intricate that one might first have thought. In this paper we have not touched on the topic of common knowledge in the social setting. Some initial observations on this are given in our [14]. Work by Ruan and Thielscher [11] and by Sano and Tojo [12] adopt a similar approach to the formalisation of 'logic in the community' [13]. Yet, although [11]

[12]Even the additional level of privacy offered here is still not perfect, as it involves some change in Mona's knowledge. She goes from knowing that Roger doesn't know that she is friends with Peggy to not knowing this. However, one may just think that privacy is a matter of degree.

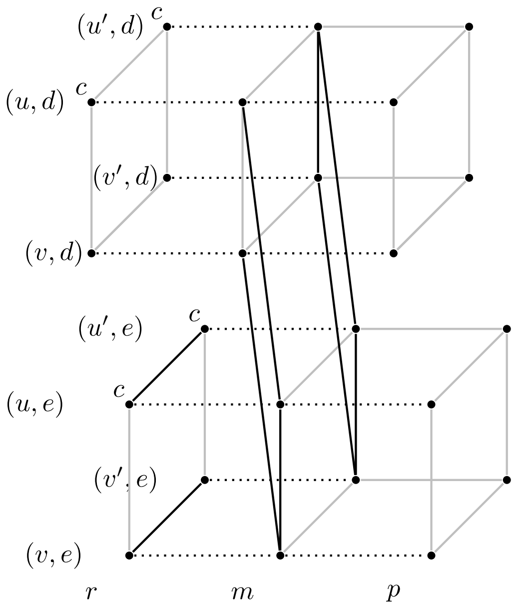



Figure 9. Peggy to Roger, privately.

includes a common knowledge operator, it captures only common knowledge of an enumerated set of agents (as in the traditional setting of multi-agent, non-social epistemic logic) and so does not capture the perspectival distinctions we have been emphasising here. The more recent [12] focusses on belief rather than knowledge, with a more limited range of dynamic operators but a more direct axiomatisation of the logic. All these approaches are compatible and point to future developments studying the interaction between propositional attitudes, indexicality and communication within social networks.

## BIBLIOGRAPHY


[1] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.

[2] Alexandru Baltag, Lawrence S. Moss, and Slawomir Solecki. The logic of public announcements, common knowledge and private suspicions. Technical Report SEN-R9922, CWI, Amsterdam, 1999.

[3] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about Knowledge*. Cambridge, MA: The MIT Press, 1995.

[4] Patrick Girard, Jeremy Seligman, and Fenrong Liu. General dynamic dynamic logic. In Thomas Bolander, Torben Braüner, Silvio Ghilardi, and Lawrence S Moss, editors, *Advances in Modal Logics Volume 9*, pages 239–260, 2012.

[5] Jaakko Hinttika. *Knowledge and Belief.* Ithaca: Cornell University Press, 1962.

[6] Fenrong Liu and Yanjing Wang. Reasoning about agent types and the hardest logic puzzle ever. *Minds and Machines*, 23(1):123–161, 2013.
[7] John-Jules Ch. Meyer and Wiebe van der Hoek. *Epistemic Logic for Computer Science and Artificial Intelligence*. Cambridge: Cambridge University Press, 1995.
[8] Eric Pacuit and Rohit Parikh. Reasoning about communication graphs. In Dov Gabbay Johan van Benthem, Benedikt Loewe, editor, *Interactive Logic*, pages 13–60. Amsterdam University Press, 2007.
[9] Jan A. Plaza. Logics of public announcements. In *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, 1989.
[10] Floris Roelofsen. Exploring logical perspectives on distributed information and its dynamics. Master's thesis, ILLC,The University of Amsterdam, 2005.
[11] Ji Ruan and Michael Thielscher. A Logic for Knowledge Flow in Social Networks. In Dianhui Wang and Mark Reynolds, editors, *AI 2011: Advances in Artificial Intelligence*, pages 511–520. Springer Berlin Heidelberg, 2011.
[12] Katsuhiko Sano and Satoshi Tojo. Dynamic epistemic logic for channel-based agent communication. In Kamal Lodaya, editor, *Logic and Its Applications*, volume 7750 of *Lecture Notes in Computer Science*, pages 109–120. Springer Berlin Heidelberg, 2013.
[13] Jeremy Seligman, Fenrong Liu, and Patrick Girard. Logic in the community. In Mohua Banerjee and Anil Seth, editors, *ICLA*, volume 6521 of *Lecture Notes in Computer Science*, pages 178–188, 2011.
[14] Jeremy Seligman, Fenrong Liu, and Patrick Girard. Facebook and the epistemic logic of friendship. In Burkhard C Schipper, editor, *Proceedings of the 14th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 229–238, Chennai, India, 2013.
[15] Johan van Benthem and Fenrong Liu. Dynamic logic of preference upgrade. *Journal of Applied Non-Classical Logic*, 17:157–182, 2007.