Logic of Promotion and Demotion

Patrick Girard

Abstract In a logic with a dimension that represents social networks, for example friendship, it is natural to add hierarchies. We can then talk about friends being better than others, and isolate best friends. However, hierarchies are not rigid: majors can become lieutenant, friendship may be strengthened or compromised, and experts can loose or gain credibility. A proper analysis of the dynamics of hierarchies is thus essential to the logic of social networks. Hierarchies of agents are structurally very similar to plausibility orders of possible worlds central to logics for belief dynamics. I use this formal analogy to show how standard policies of belief revision can be applied in social networks, thus providing systematic mechanisms of promotion and demotion in social networks.

Keywords Social networks • Dynamic logic • Belief revision • Logic in the community • Two-dimensional logic

What does promotion have to do with belief revision? Think of belief revision as dynamics over hierarchies of possible worlds. To revise with information φ is to systematically promote worlds described by φ. If you now think of φ as describing a group of agents, the φ-agents, then belief revision provides policies to systematically promote the φ-agents. Johan van Benthem (2007) describes the belief revision operations of *lexicographic upgrade* and *elite change*. About lexicographic upgrade, van Benthem says: "This move is like a social revolution where some underclass P now becomes the upper class." About elite change, he says: "Macchiavellistically,

I would like to thank Shaun White, Marcus Triplett and the anonymous referee for comments that improved the paper greatly.

P. Girard (✉)
University of Auckland, New Zealand
e-mail: p.girard@auckland.ac.nz

A. Herzig, E. Lorini (eds.), *The Cognitive Foundations of Group Attitudes and Social Interaction*, Studies in the Philosophy of Sociality 5,
DOI 10.1007/978-3-319-21732-1_5

one just co-opts the leaders of the underclass, leaving the further social order unchanged.” Transferred to a social setting, elite change and lexicographic upgrade have a literal reading instead of an analogical interpretation. This idea is at the core of the logic of promotion and demotion.

I addressed the problem of promotion and demotion in Girard (2011) and Girard and Seligman (2009) with a logic for aggregation of prioritised preference orders (cf., Andréka et al. 2002). I used a logical language with modalities $[G]\varphi$ defined over the aggregated preferences of groups of agents G. For instance, I defined the modality $[i/j]\varphi$ over the aggregation of the preferences of agents i and j by giving priority to the preferences of agent i. I then analysed promotion as a shift from a group G to a new group i/G in which agent i is given priority over other agents in G. Using this logical language, I could formalise the aggregated preferences over groups but I couldn’t reason directly about the structure of the groups.

In this paper, I will propose a logic of promotion and demotion (LPD henceforth) building on the framework of Logic in the Community (cf., Seligman et al. 2011, 2013). Logic in the community is a two-dimensional logic with epistemic and social dimensions. The social dimension contains social networks: groups of agents socially related, for example by a relation of friendship F. The modal language for this logic has a corresponding *friendship* modality $\langle F\rangle\varphi$, allowing to express social statements like “Carol is my friend” by $\langle F\rangle$**Carol**. LPD adds to this framework hierarchy relations H_a for each agent a. Hierarchies are simply total preorders over sets of friends. The language of LPD contains two modalities $\langle H_a\rangle\varphi$ and $\langle H_a^<\rangle\varphi$ defined over the hierarchy of a’s friends. You can read $\langle H_a\rangle\varphi$ as “φ holds for some friend that is at least as good as”, and $\langle H_a^<\rangle\varphi$ as “φ holds for some better friend.” For the dynamics of promotion and demotion, I use propositional dynamic logic (PDL, cf., Harel et al. 2000). As shown in Girard and Rott (2014), several belief revision policies are definable in PDL. In LPD, these are adapted to the social dimension, yielding various policies of promotion and demotion.

1 Hierarchical Models

Hierarchical models combine epistemic and social components in a two-dimensional framework. In the first dimension, possible worlds are ordered by agents according to *indistinguishability*. In the second dimension, there are two components: (1) a social network for each possible world, and (2) a hierarchy over each agent’s friends. Propositions are evaluated at world-agent pairs. So you may think of propositions as being doubly indexical: *p is true at world w for agent a*.

Given a set of propositional variables PROP and agent names AGENT, *hierarchical models* are tuples $M = \langle W, A, K, F, H, H^<, V\rangle$, in which:

- W is a non-empty set of possible worlds,
- $A = \{\underline{a}, \underline{b} \ldots\}$ is a finite set of agents,
- K is an epistemic (equivalence) relation over $W \times A$ such that $\langle (w, \underline{a}), (v, \underline{b}) \rangle \in K$ implies that $\underline{a} = \underline{b}$,
- F is a *friendship*[1] relation over $W \times A$ such that $\langle (w, \underline{o}), (v, \underline{b}) \rangle \in F$, implies that $w = v$,
- H is a collection of total preorders[2] $H_{\underline{a}}$ on the set $\{(w, \underline{b}) \in W \times A \mid \underline{a} = \underline{b}$ or $\langle (w, \underline{a}), (w, \underline{b}) \rangle \in F\}$ for every $\underline{a} \in \mathsf{AGENT}$ such that: $\langle (u, \underline{b}), (v, \underline{c}) \rangle \in H_{\underline{a}} \Rightarrow u = v$,
- $H^<$ is a collection of strict orders $H^<_a$ defined as sub-relations of H_a in the usual way[3]: $\langle (w, a), (v, b) \rangle \in H^<_a$ iff $\langle (w, a), (v, b) \rangle \in H_a$ and $\langle (v, b), (w, a) \rangle \notin H_a$, and
- V is a propositional valuation which assigns subsets of $W \times A$ to propositional variables. To each agent name $a \in \mathsf{AGENT}$, V assigns a unique agent $\underline{a} \in A$. So for $a \in \mathsf{AGENT}$, $V(a) = W \times \{\underline{a}\}$.

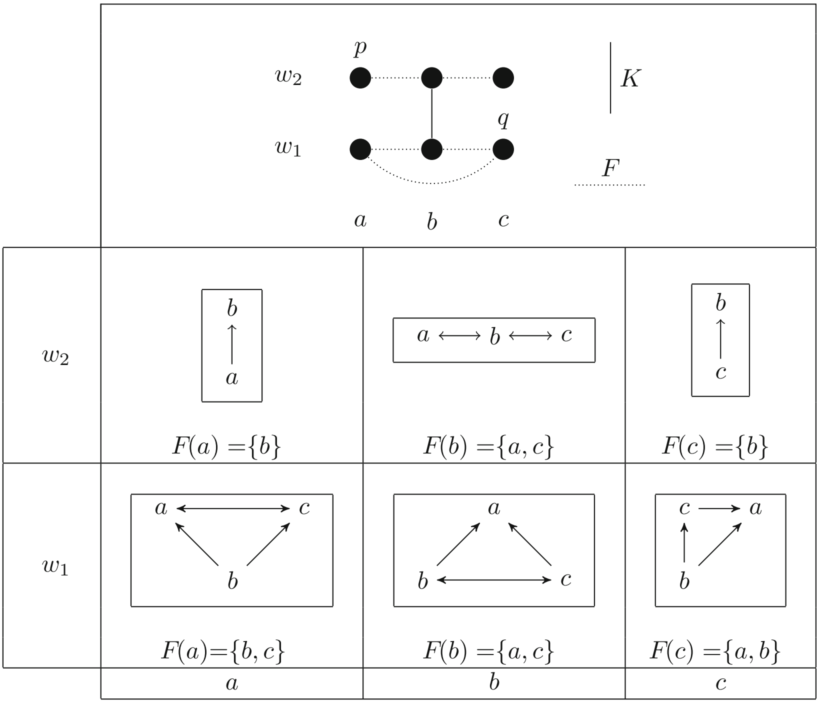
I will abuse notation and write a indiscriminately to refer to agent names $a \in \mathsf{AGENT}$ or proper agents $\underline{a} \in A$. In hierarchical models, each agent has an epistemic relation over the set of possible worlds, and each world has a friendship relation over the set of agents. The domain of a hierarchical relation H_a is the set of world-agent pairs (w, b) such that a and b are friends in world w, and hierarchies are kept world-bound. So in each world and for any two friends, agents can tell whether they are equal friends, or if one is better than the other. If $\langle (w, b), (w, c) \rangle \in H_a$, say that "$c$ is at least as good a friend to a as b". If $\langle (w, b), (w, c) \rangle \in H^<_a$, say that "$c$ is a better friend to a than b".

Example. The following represents a hierarchical model, call it M. I will refer back to M several times in the paper.

[1] I use friendship as a basic social relation for simplicity. I thus only assume F to be symmetric. Other social relations could be used, but friendship is all I need for the interpretations of promotion and demotion I have in mind.

[2] Preorders are reflexive and transitive relations. Total preorders make any two friends comparable. Friends may be equally ranked, as you should expect.

[3] Because it is defined in terms of H, $H^<$ is redundant in models. But it is not redundant in the logic, as it is well-known that strict subrelations are not modally definable. For uniformity, I thus keep $H^<$ in models.



M is a two-dimensional model with two worlds, w_1 and w_2, and three agents, *a*, *b* and *c*.[4] The top part represents the epistemic and friendship relations. For each world, there is a friendship relation represented with dotted horizontal lines. Hence, in w_1, all agents are friends together. In w_2, *b* is friends with *a* and *c*, but *a* and *c* are not friends. The vertical lines represent epistemic relations, and only agent *b* finds worlds w_1 and w_2 indistinguishable. Since *a* and *c* are friends in w_1, but not in w_2, the model depicts a situation in which agent *b* doesn't know whether *a* and *c* are friends. Finally, the proposition *p* is true at w_2 for agent *a* and *q* is true at w_1 for agent *c*. The bottom part represents every agent's hierarchy over their friends. In w_2, *b* ranks no one above others, but *a* and *c* rank *b* above themselves. In w_1, *a* ranks herself and *c* equally above *b*, *b* ranks *a* above both *b* and *c*, and finally *c* puts *a* on top of herself, with *b* at the bottom.

2 Basic Language and Semantics

Let $\rho \in \mathsf{PROP} \cup \mathsf{AGENT}$. The basis of the LPD-language for the logic of promotion and demotion is constructed from the following syntactic rules:

[4] Here and throughout the paper, I omit transitive and reflexive links whenever it improves readability in pictures.

$$\pi ::= K \mid F \mid H_a \mid H_a^<$$
$$\varphi ::= \rho \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\pi\rangle\varphi$$

The interpretation of the languages is an extension of the valuation function to a valuation $[\![\cdot]\!]^M$ assigning semantic values, or subsets of $W \times A$, to the sentences of the language.[5] In each hierarchical model $M = \langle W, A, K, F, H, H^<, V\rangle$, semantic values $[\![\varphi]\!]^M \subseteq W \times A$ and $[\![\pi]\!]^M \subseteq (W \times A)^2$ are computed in the following way:

$$\begin{array}{rcl}
[\![\rho]\!]^M & = & V(\rho), \text{ for } \rho \in \mathsf{PROP} \cup \mathsf{AGENT}. \\
[\![\neg\varphi]\!]^M & = & W \setminus [\![\varphi]\!]^M \\
[\![\varphi \wedge \psi]\!]^M & = & [\![\varphi]\!]^M \cap [\![\psi]\!]^M \\
[\![\langle\pi\rangle\varphi]\!]^M & = & \{(w,a) \in W \times A \mid \langle (w,a),(v,b)\rangle \in [\![\pi]\!]^M \;\&\; (v,b) \in [\![\varphi]\!]^M, \\
 & & \text{for some } (v,b) \in W \times A\} \\
[\![K]\!]^M & = & K \\
[\![F]\!]^M & = & F \\
[\![H_a]\!]^M & = & H_a \\
[\![H_a^<]\!]^M & = & H_a^<
\end{array}$$

Example (continuing from p. 109). Here are some formulas that are true in M.

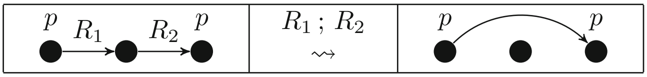
$(w_1, b) \in [\![\neg[K]\langle H_b\rangle Kq]\!]^M$	In world w_1, b doesn't know that she has a friend at least as good as herself who knows q.
$(w_2, c) \in [\![[H_c^<]\neg[K]\langle F\rangle Kp]\!]^M$	None of c's better friends know that they have a friend who knows p.
$(w_2, b) \in [\![\langle K\rangle\langle H_c^<\rangle c]\!]^M$	It is consistent with what b knows that c may rank herself above b.

3 PDL Programs

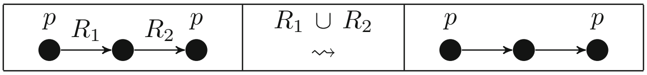
PDL-programs are tools for transforming models by redefining the relations between worlds using propositional dynamic logic (PDL). The new relations are constructed out of the old ones using PDL-*programs*. PDL-programs are built using four basic operations: *composition*, *choice*, *iteration* and *test*. From now on, I will only write ‘program’ instead of ‘PDL-program’.

[5] I choose this notation for the definition of the semantics over the more common $M, w, a \models \varphi$ for uniformity and easier integration of PDL in the next sections. In the more common notation, instead of writing $(w, a) \in [\![\langle\pi\rangle\varphi]\!]^M$, we would write $M, w, a \models \langle\pi\rangle\varphi$.

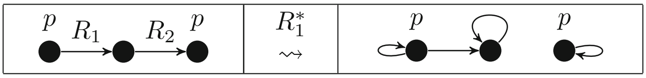
The *composition* program ‘;’ takes two relations R_1 and R_2 and combines them so that $\langle x, y\rangle \in (R_1 ; R_2)$ whenever there is a z such that R_1xz and R_2zy:



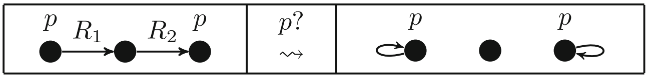
The *choice* program ‘∪’ chooses between two relations R_1 and R_2 so that $\langle x, y\rangle \in (R_1 \cup R_2)$ if either R_1xy or R_2xy:



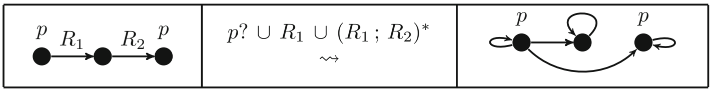
The *iteration* program ‘∗’ repeats a basic program an arbitrary finite number of times. Formally, it corresponds to taking the reflexive transitive closure of a relation, as in:



Finally, the *test* program ‘?’ tests if a formula is true at a state. As composition and choice, the test program defines a relation on models. It returns a reflexive link for worlds in which the tested formula is true[6]:



PDL can be used to define complex PDL programs. For example:



To describe programs in the language, we simply add the PDL-operators:

$$\pi ::= K \mid F \mid H_a \mid H_a^{<} \mid \pi \cup \pi \mid \pi ; \pi \mid \pi^* \mid \varphi?$$
$$\varphi ::= \rho \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\pi\rangle\varphi$$

And we expand the semantic definition accordingly:

[6]Maybe not very intuitive, but that’s how it works.

$$
\begin{aligned}
[\![\pi_1 \cup \pi_2]\!]^M &= [\![\pi_1]\!]^M \cup [\![\pi_2]\!]^M \\
[\![\pi_1 ; \pi_2]\!]^M &= \{\langle (w,a),(v,b)\rangle \mid \langle (w,a),(u,c)\rangle \in [\![\pi_1]\!]^M \\
&\quad \& \ \langle (u,c),(v,b)\rangle \in [\![\pi_2]\!]^M, \text{for some } (u,c) \in W \times A\} \\
[\![\pi^*]\!]^M &= \{\langle (w,a),(v,b)\rangle \mid (w,a) = (v,b) \text{ or } \langle (u_i,a_i),(u_{i+1},a_{i+1})\rangle \\
&\quad \in [\![\pi]\!]^M \text{ for some } n \geq 0, (u_0,a_0),\ldots (u_n,a_n) \in W \times A, \\
&\quad (u_0,a_0) = (w,a) \text{ and } (u_n,a_n) = (v,b)\} \\
[\![\varphi ?]\!]^M &= \{\langle (w,a),(w,a)\rangle \mid (w,a) \in [\![\varphi]\!]^M\}
\end{aligned}
$$

4 Promotion and Demotion

Having a social language, we can describe groups of agents with formulas. For instance, we can isolate the friends of **Barry** and **Carol** with the formula $\langle F\rangle\mathbf{Barry}\vee\langle F\rangle\mathbf{Carol}$. For any world w, any formula φ describes a group of agents, viz., the agents a such that $M, w, a \models \varphi$. Hence, we can use belief revision operations on φ to promote or demote groups of agents. I will use the following abbreviations:

$$
\begin{aligned}
H_a^{\varphi} &::= (\varphi ? \,;\, H_a \,;\, \varphi ?) \\
\mathbf{best_a}(\boldsymbol{\varphi}) &::= \langle F\rangle a \wedge \varphi \wedge \neg\langle H_a^{<}\rangle\varphi
\end{aligned}
$$

For any formula φ, H_a^{φ} restricts a's hierarchy to agents described by φ and $\mathbf{best_a}(\boldsymbol{\varphi})$ isolates the best φ-agents in a's hierarchy. For example, take $\varphi = \langle F\rangle a$, i.e., agents satisfying the formula which says that a is amongst their friends, then $\mathbf{best_a}(\boldsymbol{\varphi})$ returns a's best friends. Or one can think of φ as ascribing expertise to agents, so that promoting φ-agents is giving priority to φ-experts.

I first consider two operations of promotion which I call, following the terminology of Girard and Rott (2014) and Rott (2009), *conservative* and *moderate*. Conservative promotion promotes the best φ-agents on top of the hierarchy and preserves the ranking otherwise:

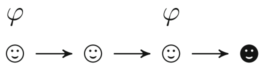
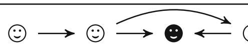
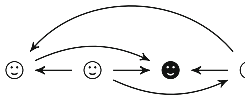
Conservative Promotion
$\mathsf{CProm_a}(\varphi) \;=\; H_a^{\neg\mathbf{best_a}(\boldsymbol{\varphi})} \cup ((\langle F\rangle a \wedge \neg\mathbf{best_a}(\varphi))? \,;\, F^* ;\, \mathbf{best_a}(\varphi)?)$

Notice the role of F^* to ensure that all of a's friends can be accessed, creating (possibly) new links ranking a's best φ-friends over the others. Since F is a symmetric relation, F^* is an equivalence relation (it takes the reflexive transitive closer of a symmetric relation). Whenever I need to access all of a's friends in programs, I use F^* in a similar fashion.

Moderate promotion acts like conservative promotion, but promotes *all* φ-agents instead of only the best ones:

Moderate promotion
$\mathsf{MProm_a}(\varphi) \quad = \quad H_a^{\varphi} \cup H_a^{\neg\varphi} \cup ((\langle F\rangle a \wedge \neg\varphi)?\,;\, F^*\,;\, (\langle F\rangle a \wedge \varphi)?)$

As a simple representation, here's the result of applying conservative and moderate promotion to the same initial model:

Conservative and Moderate Promotion		
	$\mathsf{CProm_a}(\varphi)$	
	$\mathsf{MProm_a}(\varphi)$	

The black figures represent best friends. The three operations of promotion agree on who should be the best friends after promoting φ agents, but they disagree on how to order the remaining friends. Conservative promotion preserves most of the initial hierarchy, only taking the best φ-agents and putting them on top. Moderate promotion reorders every agent, by putting all φ-agents over all $\neg\varphi$-agents.

For demotion, I also define a conservative and a moderate version. As these operations are based on doxastic operations with a minimalist attitude, the result of demoting φ-agents doesn't entail that φ-agents are no longer best friends. What demotion does to a group is to make sure that the set of best friends is no longer only constituted by φ-agents.

Conservative demotion takes the best $\neg\varphi$-agents and puts them on a par with other best friends, but preserves the hierarchy otherwise.

Conservative Demotion
$\mathsf{CDem_a}(\varphi) \quad = \quad H_a^{\neg\mathbf{best_a}(\neg\varphi)} \cup (F^*\,;\, \mathbf{best_a}(\neg\varphi)?) \cup (F^*\,;\, \mathbf{best_a}(\top)?)$

Conservative demotion guarantees that the ruling class no longer consists only of φ-agents.

Moderate demotion again preserves best φ-friends, but it puts all other φ-agents under $\neg\varphi$-agents:

Moderate Demotion
$\mathsf{MDem_a}(\varphi) \quad = \quad H_a^{\varphi} \cup H_a^{\neg\varphi}$ $\cup\, ((\langle F\rangle a \wedge \varphi \wedge \neg\mathbf{best_a}(\top))?\,;\, F^*\,;\, (\langle F\rangle a \wedge \neg\varphi)?)$ $\cup\, (F\,;\, \mathbf{best_a}(\neg\varphi)?) \cup (F^*\,;\, \mathbf{best_a}(\top)?)$

The following diagram illustrates the difference between conservative and moderate demotion. As was the case with promotion, the two operations agree on who become the best friends after demotion, but diverge in how they treat other agents.

Conservative and Moderate Demotion		
φ φ ☺ ⟶ ☺ ⟶ ☺ ⟶ ☻	$\mathsf{CDem_a}(\varphi)$	☺ ⟶ ☺ ⟶ ☻ ⟷ ☻
	$\mathsf{MDem_a}(\varphi)$	☺ ⟵ ☺ ⟶ ☻ ⟷ ☻

5 PDL-Transformations

The final installments in the logic of promotion and demotion are PDL transformations, taken from Girard et al. (2012).[7] PDL-transformations are collections of PDL programs that operate *in parallel*. A PDL-transformation Λ is a collection of programs $\Lambda(K)$, $\Lambda(F)$ and $\Lambda(H_a)$ that redefines each of the relations. I represent PDL-transformations in the following way:

Λ		
K	$:=$	$\Lambda(K)$
F	$:=$	$\Lambda(F)$
H_a	$:=$	$\Lambda(H_a)$, for every $a \in \mathsf{AGENT}$

A PDL-transformation is thus a way of combining several programs to redefine the relations of a model. From now on, I will just write 'transformation' instead of 'PDL-transformation'.

Let Λ be a transformation and let $M = \langle W, A, K, F, H, H^<, V\rangle$ be a hierarchical model. $\Lambda(M) = \langle W, A, \Lambda(K), \Lambda(F), \Lambda(H_a), \Lambda(H_a^<), V\rangle$ is a new hierarchical model resulting from applying Λ to M, in which:

$$\begin{array}{lcl} \Lambda(K) & = & [\![\Lambda(K)]\!]^M \\ \Lambda(F) & = & [\![\Lambda(F)]\!]^M \\ \Lambda(H_a) & = & [\![\Lambda(H_a)]\!]^M \\ \Lambda(H_a^<) & = & [\![(\Lambda(H_a))^<]\!]^M \end{array}$$

In some cases, transformations preserve some relations in the model exactly as they were. For instance, in the logic of promotion and demotion, they never affect

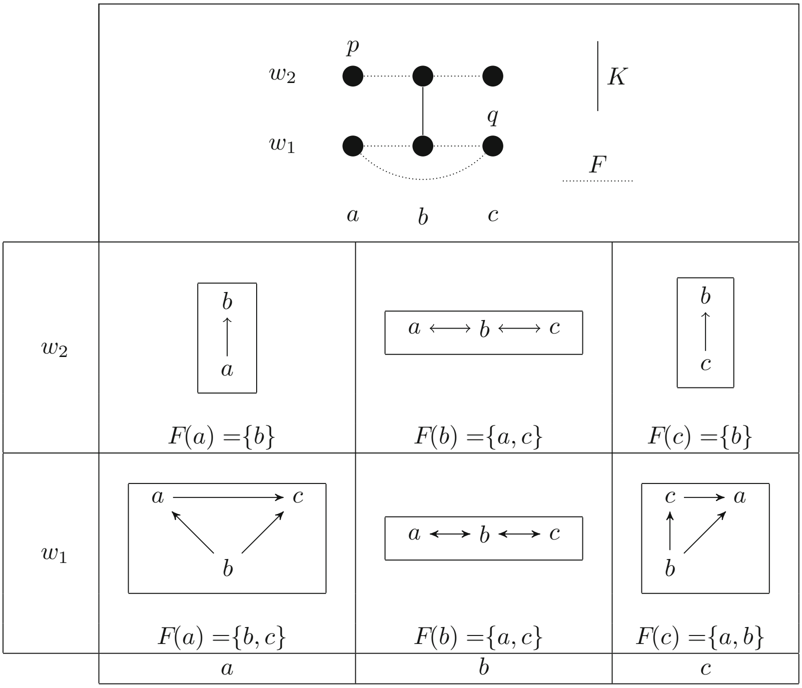
[7]For the details of the general case of PDL-transformations, the reader should consult section 1 of Girard et al. (2012). I give here a self-contained special case of PDL-transformations required for my purposes.

the epistemic relation. I will thus shorten the representation of transformations by omitting relations that are preserved, as in:

β		
H_a	$:=$	$\mathsf{MProm_a}(c)$
H_b	$:=$	$\mathsf{CDem_b}(a)$

The transformation β contains two programs, one for a and one for b. With β, a moderately promotes c and his friends, and b conservatively demotes a. All other relations are not affected by β, and so are omitted from the representation.

Example (continuing from p. 111). Let's see how β operates on M by computing $\beta(M)$:

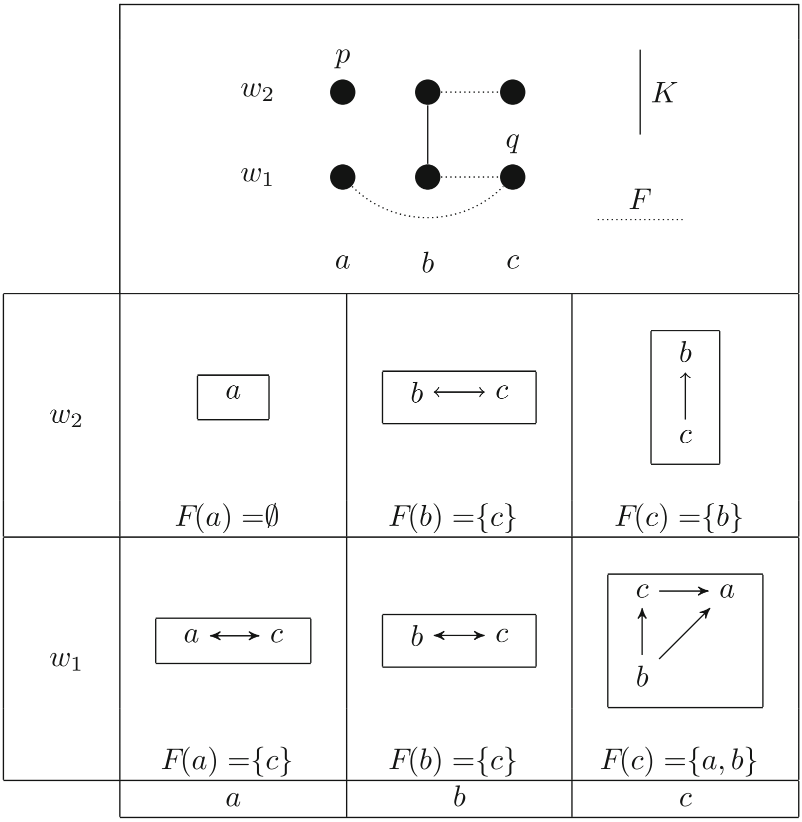


Transformations can do more than simply combining social action for all agents, as in the simple example above. They can also define actions of promotion and demotion that are not reducible to simple programs. As an example, here's a transformation for *radical* promotion that operates on both the friendship and the hierarchy relation to define a new hierarchy. Radical promotion is an operation by which an agent breaks the relationship with some of her friends, but keeps the hierarchy amongst the remaining friends as it was before. Since friendship is a symmetric relation, a breaking the relationship with b forces b to also reconfigure her hierarchy:

Radical Promotion – $\mathsf{RProm}_{\mathsf{a}}(\varphi)$		
F	$:=$	$F^{\neg a} \cup (a?\,;\,F\,;\,\varphi?) \cup ((\neg a \wedge \varphi)?\,;\,F\,;\,a?)$
H_a	$:=$	$H_a^{\varphi} \cup (a?\,;\,H_a\,;\,\varphi?) \cup (\varphi?\,;\,H_a\,;\,a?)$
H_i	$:=$	$(\varphi?\,;\,H_i^{\neg a}) \cup (\neg\varphi?\,;\,H_i)$ for $i \neq a, i \in \mathsf{AGENT}$

This definition is tailored to the friendship relation F being a symmetric relation. When a drops φ-agents amongst her friends, those agents are no longer friends with a and must adapt their hierarchy relations accordingly. H_a is transformed so that a only ranks herself and φ-agents just as she used to rank them. Finally, all other agents, if they are φ-agents, have to exclude a from their hierarchies, as they are no longer friends.

Example (continuing from p. 111). As an example of a transformation acting on a model, let's compute the result $\mathsf{RProm}_{\mathsf{b}}(c)(M)$ of b radically promoting c in model M:



6 Logic of Promotion and Demotion: LPD

To describe transformations in the language of LPD, we add modalities $\langle\Lambda\rangle$ for each acceptable transformation Λ[8]:

$$\pi ::= K \mid F \mid H_a \mid H_a^< \mid \pi \cup \pi \mid \pi ; \pi \mid \pi^* \mid \varphi?$$
$$\varphi ::= \rho \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\pi\rangle\varphi \mid \langle\Lambda\rangle\varphi$$

We finally expand the semantic definition for the transformation modalities:

$$[\![\langle\Lambda\rangle\varphi]\!]^M \quad = \quad [\![\varphi]\!]^{\Lambda(M)}$$

An accustomed reader or a keen logician might now expect me to axiomatise LPD and prove completeness. I will not do so in this paper. As is common in the dynamic epistemic logic literature, completeness for dynamics is not a difficult technical problem, because it can be avoided. In the case of LPD, we can use a translation φ^Λ of formulas of the LPD language with transformation modalities to formulas without them, so that:

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

Whereas a transformation Λ operates on a model M to create a new model $\Lambda(M)$, the translation of formulas encodes the result of the transformation in the language without transformation modalities. It's as though the static language could predict what will happen after models are transformed.

The proof strategy I used is directly borrowed from the GDDL logic of Girard et al. (2012) and is straightforwardly adapted to the LPD context. The translation φ^Λ needed for the reduction is the following:

$$\begin{array}{lll@{\qquad}lll}
p^\Lambda & = & p & K^\Lambda & = & \Lambda(K) \\
(\neg\varphi)^\Lambda & = & \neg\varphi^\Lambda & F^\Lambda & = & \Lambda(F) \\
(\varphi\wedge\psi)^\Lambda & = & (\varphi^\Lambda\wedge\psi^\Lambda) & H_a^\Lambda & = & \Lambda(H_a) \\
(\langle\pi\rangle\varphi)^\Lambda & = & \langle\pi^\Lambda\rangle\varphi^\Lambda & (H_a^<)^\Lambda & = & (H_a^\Lambda)^< \\
 & & & (\pi_1 \cup \pi_2)^\Lambda & = & \pi_1^\Lambda \cup \pi_2^\Lambda \\
 & & & (\pi_1; \pi_2)^\Lambda & = & \pi_1^\Lambda; \pi_2^\Lambda \\
 & & & (\pi^*)^\Lambda & = & (\pi^\Lambda)^* \\
 & & & (\varphi?)^\Lambda & = & (\varphi^\Lambda)?
\end{array}$$

With this translation, a straightforward induction establishes Lemma 1, which states that the logic with transformations can be systematically reduced to one without them:

[8]We only accept transformations that produce hierarchical models. Here's a technical problem for the inclined reader: how do you characterise acceptable transformations for different logics? That is, if I give you a class of models, how can you isolate transformations that will produce models within the same class?

Lemma 1. *For each world-agent pair* (w, a) *of* $\Lambda(M)$ *and* (v, b) *of* M*, and for each formula* φ*:*

$$\begin{array}{lll} (w,a) \in [\![\varphi^{\Lambda}]\!]^{M} & \textit{iff} & (w,a) \in [\![\varphi]\!]^{\Lambda(M)} \\ \langle (w,a),(v,b)\rangle \in [\![\pi^{\Lambda}]\!]^{M} & \textit{iff} & \langle (w,a),(v,b)\rangle \in [\![\pi]\!]^{\Lambda(M)} \end{array}$$

Therefore, as far as completeness of the logic is concerned, no additional work is required to axiomatise $\langle\Lambda\rangle$ modalities.

7 Conclusion

This concludes our investigation of promotion and demotion as can be expressed in LPD. Many more operations can be defined in LPD, but I have selected those which I think are most interesting. I have left some topics untouched in this paper. In particular, I haven't mentioned anything about the axiomatisation of the static part of LPD. Although not a trivial task, I believe this will not present serious difficulties. The axiomatisation of the hierarchical modalities would be based on that for total preorders for $[H_a]\varphi$ with an axiom for the proper interaction with $[H_a^<]$: $a \rightarrow (H_b^< \varphi \leftrightarrow H_b(\varphi \wedge \neg H_b a))$. One also needs an axiom for the proper interaction with the friendship modality: $[F]\varphi \rightarrow [H_a]\varphi$. Another aspect of the GDDL which I haven't exploited is the formalisation of *private* actions, in which agents secretly change the hierarchy of their friends with some of them being ignorant of the change.

I have only considered operations of promotion and demotion on groups as they were suggested by the doxastic operation of revision and contraction found in the AGM literature. But the LPD language is very flexible, and we could use it to formalise a range of different notions of promotion and demotion. One could for instance define operations in which a demotion of φ-agents would put all φ-agents under all $\neg\varphi$-agents, or would put all best φ-agents under all best $\neg\varphi$-agents; with neither of these alternative definitions would best φ-agents remain best friends after the demotion, as is the case when we use my own definitions. The preliminary framework and results I have provided encourage further investigation in a number of different directions.

References

Andréka, Hajnal, Mark Ryan, and Pierre-Yves Schobbens. 2002. Operators and laws for combining preference relations. *Journal of Logic and Computation* 12(1): 13–53.

Girard, Patrick. 2011. Modal logic for lexicographic preference aggregation. In *Games, norms and reasons*, 97–117. Dordrecht: Springer.

Girard, Patrick, and Hans Rott. 2014. Belief revision and dynamic logic. In *Trends in logic, outstanding contributions: Johan F. A. K. Van Benthem on logical and informational dynamics*, vol. 5, ed. Alexandru Baltag and Sonja Smets. Cham: Springer.

Girard, Patrick, and Jeremy Seligman. 2009. An analytic logic of aggregation. In *Logic and its applications*, Lecture notes in computer science, vol. 5378, ed. R. Ramanujam and Sundar Sarukkai, 146–161. Berlin/Heidelberg: Springer.

Girard, Patrick, Jeremy Seligman, and Fenrong Liu. 2012. General dynamic dynamic logic. *Advances in Modal Logics* 9: 239–260

Harel, David, Jerzy Tiuryn, and Dexter Kozen. 2000. *Dynamic logic*. Cambridge: MIT.

Rott, Hans. 2009. Shifting priorities: Simple representations for twenty-seven iterated theory change operators. In *Towards mathematical philosophy*, Trends in logic, vol. 28, ed. David Makinson, Jacek Malinowski, and Heinrich Wansing, 269–296. Dordrecht: Springer.

Seligman, Jeremy, Fenrong Liu, and Patrick Girard. 2011. Logic in the community. In *Logic and Its Applications*, Lecture notes in computer science, vol. 6521, ed. Mohua Banerjee and Anil Seth, 178–188. Berlin/Heidelberg: Springer.

Seligman, Jeremy, Fenrong Liu, and Patrick Girard. 2013. Facebook and the epistemic logic of friendship. In *TARK XIV: Proceedings of the 13th conference on theoretical aspects of rationality and knowledge*, Chennai: India.

van Benthem, Johan. 2007. Dynamic logic for belief revision. *Journal of Applied Non-classical Logic* 17(2): 129–155.