Local Belief Dynamics in Network Knowledge Bases

FABIO R. GALLO and GERARDO I. SIMARI, Department of Computer Science and Engineering, Universidad Nacional del Sur (UNS) and Institute for Computer Science and Engineering (UNS-CONICET), Argentina
MARIA VANINA MARTINEZ, Department of Computer Science, Universidad de Buenos Aires (UBA) and Institute for Computer Science Research (UBA-CONICET), Argentina
NATALIA ABAD SANTOS, Department of Mathematics, Universidad Nacional del Sur (UNS), Argentina
MARCELO A. FALAPPA, Department of Computer Science and Engineering, Universidad Nacional del Sur (UNS) and Institute for Computer Science and Engineering (UNS-CONICET), Argentina

People are becoming increasingly more connected to each other as social networks continue to grow both in number and variety, and this is true for autonomous software agents as well. Taking them as a collection, such social platforms can be seen as one complex network with many different types of relations, different degrees of strength for each relation, and a wide range of information on each node. In this context, social media posts made by users are reflections of the content of their own individual (or local) knowledge bases; modeling how knowledge flows over the network—or how this can possibly occur—is therefore of great interest from a knowledge representation and reasoning perspective. In this article, we provide a formal introduction to the *network knowledge base* model, and then focus on the problem of how a single agent's knowledge base changes when exposed to a stream of news items coming from other members of the network. We do so by taking the classical belief revision approach of first proposing desirable properties for how such a local operation should be carried out (theoretical characterization), arriving at three different families of local operators, exploring concrete algorithms (algorithmic characterization) for two of the families, and proving properties about the relationship between the two characterizations (representation theorem). One of the most important differences between our approach and the classical models of belief revision is that in our case the input is more complex, containing additional information about each piece of information.

CCS Concepts: • **Information systems → Social networking sites**; **Collaborative and social computing systems and tools**; • **Computing methodologies → Reasoning about belief and knowledge;**

Additional Key Words and Phrases: Network knowledge bases, belief revision, social networks

4

This work was funded in part by Universidad Nacional del Sur (UNS) under grants PGI 24/N046 and PGI 24/ZN34, Secretaría de Investigación Científica y Tecnológica FCEN–UBA (RESCS-2020-345-E-UBA-REC), CONICET under the PIP grants, and Agencia Nacional de Promoción Científica y Tecnológica, Argentina under grants PICT-2018-0475 (PRH-2014-0007) and PICT-2016-0215.

Authors' addresses: F. R. Gallo, G. I. Simari, and M. A. Falappa, Department of Computer Science and Engineering, Universidad Nacional del Sur (UNS) and Institute for Computer Science and Engineering (UNS-CONICET), San Andres 800, (8000) Bahia Blanca, Buenos Aires, Argentina; emails: {fabio.gallo, gis, mfalappa}@cs.uns.edu.ar; M. V. Martinez, Department of Computer Science, Universidad de Buenos Aires (UBA) and Institute for Computer Science Research (UBA-CONICET), Intendente Güiraldes, Acceso Pabellón 1 2160, (1428) Buenos Aires; email: mvmartinez@dc.uba.ar; N. A. Santos, Department of Mathematics, Universidad Nacional del Sur (UNS), Av. Alem 1253, (8000) Bahia Blanca, Buenos Aires, Argentina.

1529-3785/2021/10-ART4 $15.00
https://doi.org/10.1145/3477394

ACM Reference format:
Fabio R. Gallo, Gerardo I. Simari, Maria Vanina Martinez, Natalia Abad Santos, and Marcelo A. Falappa. 2021. Local Belief Dynamics in Network Knowledge Bases. *ACM Trans. Comput. Logic* 23, 1, Article 4 (October 2021), 36 pages.
https://doi.org/10.1145/3477394

1 INTRODUCTION

Online platforms for sharing content with both friends and perfect strangers, commonly known today as *social media*, now dominate many aspects of communications among people, between a person and their government, as well as between companies and the people who consume their products and services. One of the most important aspects of this is how *information flow* has changed with respect to how things were less than two decades ago—television, radio, and printed media have been largely replaced with content on a wide variety of social media platforms such as Twitter, Facebook, Instagram, YouTube, Weibo, Quora, Pinterest, and LinkedIn—among many others—where information ranges from simple text comments and "likes" or "up/down votes," to multimedia content such as audio, photos, and video. The 2016 presidential elections in the US and the Brexit vote in the UK are clear examples of how such content can have an influence on people's opinions and beliefs and, through this, on world-changing events [40, 45, 48]. The recent surge in so-called *fake news* is another reminder of the power of these platforms [2, 56].

In this article, we address the broad issue of understanding how information is shared by people (or in general autonomous agents) that are part of one or more social networks. Since there are a wide variety of such networks, and most people are active on at least two, we developed a model that is capable of representing the salient aspects of a wide variety of data and communication structures: Nodes represent agents and can be labeled with different pieces of information about them (such as the date that they joined specific platforms, city of residence, age, etc.), while edges represent their relationships with other agents and can also be labeled with relevant information (for instance, nature and strength of the relationship); finally, each node has a local knowledge base representing the beliefs currently held by each entity. One of the key advantages of this so-called **Network Knowledge Base (NKB)** model [22, 25] is its ability to seamlessly integrate different data sources into a single model. This underlying idea of combining sources into one network model is not new—several such models have been proposed in the wide-reaching social networks literature, which includes contributions from many different disciplines like psychology, sociology, biology, computer science, and physics; they can in general be grouped under the name *multilayer* or *multiplex* networks—cf. Reference [37] for a recent survey of the most prominent models.

However, the problem of studying the design and implementation of principled operations for characterizing the flow of knowledge in these generalized social networks has only recently started to be addressed [23, 25]. Here, we continue this line of research by taking the classical belief revision approach [1, 33] of proposing a set of theoretical properties, called *postulates*, of how such operations should be carried out, proposing concrete algorithms to implement them in practice, and studying how the two approaches are related. Though an experimental evaluation of our theoretical and algorithmic characterizations is outside the scope of this article, we refer the reader to Reference [25] for some preliminary results obtained using our NKB model to characterize *user types* in Twitter based on the tone adopted in their posts in relation to the tone observed in their individual feeds. Another recent work focused on extending this to the application of our model in conjunction with machine learning classifiers to predict user reactions to the content of their feeds [24].

The following is an example of an application of our Network Knowledge Base model.

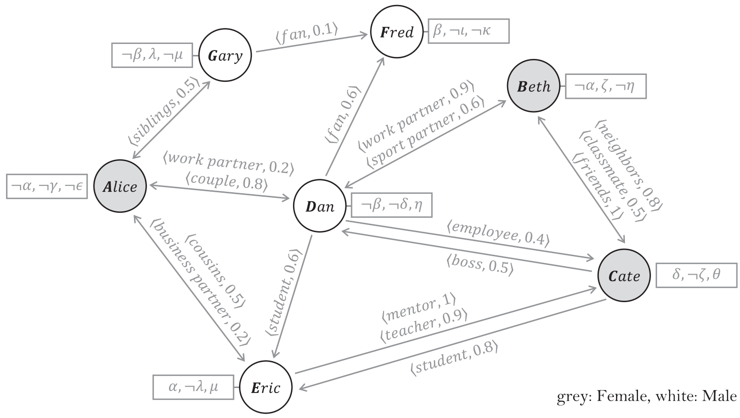

Fig. 1. Example of a *Network Knowledge Base*—a generalized social network with different kinds of weighted relations and knowledge bases associated with each node.

Example 1.1 (Integration of Social Media Platforms). We now describe a setting that we use as a running example to motivate and illustrate the presentation. Consider a social media platform like Facebook, Weibo, Instagram, or Twitter, where people are connected to each other and share thoughts and ideas in the form of text comments or visual content such as photos and videos. We not only wish to model that connections exist between users but also the different levels of *strength* or *weight* of these relations. Furthermore, we would like to model the information that is available for each user, such as age, gender, current city, and so on.

Figure 1 shows a network consisting of seven users; their relationships are represented with labeled edges—in general multiple relationships can exist between two people, e.g., *Dan* and *Beth* are *work* and *sports partners*; these labels can of course also represent follow/friend relationships on different social platforms. Labels indicating how close they are as a value in the real unit interval are called *weights*; these can be automatically calculated (as discussed below). Labels associated with nodes (not included in this example for simplicity of presentation) can represent their attributes, such as age, place of work, or the date on which they joined a social platform. Finally, each vertex has associated a local *set of beliefs*, which we refer to as the agent's *local knowledge base*. ■

A note on weighted edges. There are many ways in which weights can be automatically calculated based on user activity on social platforms. For instance, for an edge from user u to user v labeled *Facebook friend*, it could be based on easy-to-compute[1] metrics such as the proportion of posts by v that involve reactions by u, the frequency with which u initiates a private conversation with v, how many groups they both belong to, or more explicit signals such as whether or not u has marked v as a favorite or close friend. Such metrics can also be refined by taking into account further aspects, such as how long it typically takes u to react to posts by v, the kind of reaction, the average comment length, and so on. Assigning weights to relationships has also been previously considered in the multi-layer/multiplex networks literature; for example, Reference [41] studies how the strength of ties can be calculated from Facebook data, Reference [46] studies weighted multiplex networks, focusing on empirical evidence of edge weights in a dataset of articles published in American Physical Society (APS) journals, and Reference [6] illustrates how this can be done in another scientific collaboration dataset as well as in a dataset describing connections between airports.

[1] Assuming access to the relevant data is available.

In this setting, posts made by users are seen by all their connections; when this occurs, we assume that each user adopts a position regarding the information that they receive from their friends. For instance, it could be seen but ignored, or perhaps an evaluation is made to see if the knowledge has enough epistemic value to be integrated into their own knowledge base. Users therefore generally have many possible sources of information that can be both mutually inconsistent (e.g., friend A stating "α" and friend B "$\neg\alpha$"), and/or posts that contradict their local belief base (e.g., friends A and B stating "$\neg\alpha$" when the user in question's local knowledge base currently contains α). In such cases, it would be natural for users to resort to the closeness of the friend who made the post, how many friends share the opinion, and so on, to make up their minds regarding the content in question.

Given the expressive power of the NKB model, there are plenty of domains in which this model could be useful. For instance, in the problem of selecting the best users to be included in a marketing campaign based on social media; users who have the greatest number of connections are not always the best candidates for being chosen as "seeds" of online marketing campaigns [14]. In this scenario, our model could provide a broad view for identifying the potentially most influential users according to the current state of their knowledge bases and the contagion effect that can be leveraged to reach many users in the network. Another possible application of this model is as a support tool—in combination with NLP and ML tools, among others—for detecting unfair competition, for instance via *sock puppets*[2]; in this setting, NKBs could be useful in detecting information flow patterns that are typical of sock puppet campaigns.

The main contributions of this article are the following:

- Formal introduction of the Network Knowledge Base model and the associated global and local belief revision problems.
- Introduction of a set of desirable properties for local belief revision operations (*postulates*).
- Theoretical characterization of three families of local belief revision operators: *restrained*, *weakly restrained*, and *social*.
- Algorithmic characterization of the *restrained* and *weakly restrained* families of local operators.
- Representation theorem relating the proposed theoretical and algorithmic characterizations.
- Time and space complexity analysis of our proposed algorithm for implementing restrained and weakly restrained operators.

These contributions appear in the rest of this article as follows: Section 2 formally presents the concept of NKBs; then, Section 3 discusses the general global and local belief revision processes that can occur in NKBs as agents react to content posted by others, focusing especially on the local revision operators that govern how each agent changes its own knowledge base—we present a set of postulates and characterize three general families of local operators. Section 4 discusses the construction (*algorithmic characterization*) of *restrained* and *weakly restrained* operators, and presents our main representation theorem. Finally, Sections 5 and 6 present related work and conclusions, respectively. Additional material can be found in Appendix A.

2 NETWORK KNOWLEDGE BASES

In this section, we provide a formal introduction to the concept of NKBs; this model was first briefly proposed in Reference [23] and also appeared in a less formal presentation in Reference [25].

[2] This term refers to a false identity assumed by a member of an Internet community who speaks to, or about, themselves while pretending to be another person.

We assume a language $\mathcal{L}$ built from a finite set of propositional symbols *Pred*, in which the only connective is $\neg$; *therefore, all elements in our language are ground literals.* We say that two literals l_1, l_2 are contradictory if and only if $l_1 \equiv \neg l_2$. We also have two arbitrary but fixed disjoint sets *VP*, *EP* of *vertex* and *edge* n-ary predicate symbols (with $n \geq 0$), respectively, and a finite set of constants Δ. We use $\mathcal{L}_V$ (respectively, $\mathcal{L}_E$) to denote ground atoms over *VP* (respectively, *EP*) built from *VP* (respectively, *EP*) and Δ. We first recall the definition of a social network, presenting a slight variation of the one in Reference [55].

Definition 2.1. A *Social network* is a 4-tuple $(V, E, l_{vert}, l_{edge})$, where

(1) V is a finite set whose elements are called *vertices.*
(2) $E \subseteq V \times V$ is a finite set whose elements are called *edges.*
(3) $l_{vert} : V \rightarrow 2^{\mathcal{L}_V}$ is a function called a *vertex labeling function.*
(4) $l_{edge} : E \rightarrow 2^{\mathcal{T}}$ is a function called an *edge labeling function*, where $\mathcal{T} = \{\langle b, w\rangle \mid b \in \mathcal{L}_E, w \in [0, 1]\}$.

Note that this definition of social network allows for multiple relations to exist between the same pair of nodes, since multiple labels can be attached to the same edge. Also, multiple labels can be attached to a vertex. Multiple relation labels allow to model the fact that two vertices, for instance, are both neighbors and classmates, while multiple vertex labels allow to represent user data such as age, nationality, place of work, and so on. A *network knowledge base* is then simply a social network in which a set of propositional literals is associated with each vertex.

Definition 2.2. A *Network KB* (NKB for short) is a 5-tuple $(V, E, l_{vert}, l_{edge}, K)$, where the first four elements comprise a social network, and $K : V \rightarrow 2^{\mathcal{L}}$ is a mapping assigning a knowledge base to each vertex. Given v, $K(v)$ is called the *knowledge base* associated with vertex v.

Additionally, we further enrich the notion of network with a set of constraints that conditions (and relates) both the structural part of the network (i.e., its relationships) and the **knowledge bases** (**KBs**, for short) of their users.

Definition 2.3. A *constraint* C over an NKB $(V, E, l_{vert}, l_{edge}, K)$ is a pair (S, B) where, given $\{v_1, \ldots, v_n\} \subseteq V$, and $\{e_1, \ldots, e_m\} \subseteq E \cap \{v_1, \ldots, v_n\} \times \{v_1, \ldots, v_n\}$,

(1) S, called the *structural part*, contains a Boolean combination (i.e., using $\wedge$, $\vee$, and $\neg$) of conditions that can be of either of the following forms:
  - $(a \in l_{vert}(v))$, with $a \in \mathcal{L}_V$ and $v \in \{v_1, \ldots, v_n\}$;
  - $(\langle b, w\rangle \in l_{edge}(e))$, with $\langle b, w\rangle \in \mathcal{T}$, where $\mathcal{T} = \{\langle b, w\rangle \mid b \in \mathcal{L}_E, w \in [0, 1]\}$.
(2) B is called the *belief part* and contains a Boolean combination (i.e., using $\wedge$, $\vee$, and $\neg$) of statements of the form either $(c \in K(v))$, with $c \in \mathcal{L}$ and $v \in V$.

Satisfaction and Consistency. Given a constraint $C = (S, B)$, we say that a set of vertices $\{v_1, \ldots, v_n\}$ and edges $\{e_1, \ldots, e_m\}$ *satisfy* S if all the conditions of the formula hold when we replace the vertices and the edges appropriately. Similarly, we say that a set of vertices $\{v_1, \ldots, v_n\}$ *satisfies* B if there exists a way to replace the vertices such that the formula holds. Finally, an NKB *satisfies* C if for every subset of vertices and edges that satisfy S, B is also satisfied. Given this semantics, it may also be convenient to write constraints as $S \rightarrow B$.

Example 2.1. Let v_1, $v_2 \in V$, and $e(v_1, v_2), e(v_2, v_1) \in E$; we have a constraint $C_1 = (S_1, B_1)$ defined by:

- S_1: $\langle employee, w_1\rangle \in l_{edge}(v_1, v_2) \wedge \langle boss, w_2\rangle \in l_{edge}(v_2, v_1)$;
- B_1: $\Big((\delta \in K(v_1)) \wedge (\delta \in K(v_2))\Big) \vee \Big((\neg\delta \in K(v_1)) \wedge (\neg\delta \in K(v_2))\Big)$.

Constraint C_1 says that if there is an edge labeled with *employee* between two vertices and another edge in the opposite direction labeled with *boss* (formula S_1), then the knowledge bases for these vertices should either both be positive or both negative with respect to proposition δ (formula B_1). In Figure 1 we can see that although S_1 holds, B_1 does not, since δ belongs to Cate's KB and $\neg\delta$ belongs to Dan's—C_1 is therefore not satisfied. ∎

We are now ready to define consistency of an NKB.

Definition 2.4. Given a finite set of constraints *IC*, we say that an NKB $= (V, E, l_{vert}, l_{edge}, K)$ is *consistent* w.r.t. *IC* if it satisfies all constraints in *IC* and for every $v \in V$, $K(v)$ does not contain contradictory literals.

The Network Belief Dynamics Problem. Consider an epistemic input comprised of a set of decisions coming from the neighbor nodes about how their beliefs are changing; we call such decisions *news items*:

Definition 2.5. A *news item* consists of a triple (o, l, d), where $o \in V$ is the *origin*, $l \in \mathcal{L}$ is a literal, and d is an indication of its new status in the origin node's belief base: a (l was *added* to $K(o)$) or r (l was *removed* from $K(o)$).

We assume that news items are only created when they are not redundant—therefore, for (o, l, a) we have that $l \notin K(o)$, and for (o, l, r) we have that $l \in K(o)$ (in both cases before the operation). We also point out that news items should not be interpreted as facts, but rather as statements made by actors in social platforms who may have different motivations. News items are thus part of an epistemic input (as discussed in the next section) that merely characterizes what users are posting (which may or may not accurately reflect their beliefs).

Definition 2.6. A set of news items P is *consistent* if there do not exist $p_1 = (o_1, l_1, d_1)$ and $p_2 = (o_2, l_2, d_2)$ such that:

(i) $o_1 = o_2$, $l_1 = l_2$, and $d_1 \neq d_2$.
(ii) $o_1 = o_2$, $l_1 \equiv \neg l_2$, and either
- $d_1 = d_2 = a$, or
- $d_1 = d_2 = r$.

For instance, $P_1 = \{(u, \alpha, a), (u, \alpha, r)\}$ is inconsistent (case i), since the same user is both adding and removing the same literal.

In the rest of the article, we assume all sets of news items to be consistent. This constraint could be eliminated by admitting any set of items and assuming the existence of a *filtering function* as in operators of selective revision [20] or a condition on the set of credible sentences in operators of credibility limited revision [34]. Another simplifying assumption that we make here is that we can derive news items from social media posts; one way in which this can be tackled is via the guided application of machine learning techniques for natural language processing tasks (cf. Reference [64] for a survey of such approaches). For instance, in the posts illustrated in Figure 2, the specific dimensions (price, parking, veggie, etc.) could be fixed beforehand, and automated **Natural Language Processing (NLP)** tools applied to posts to determine if they are relevant to each one.

The following example illustrates how a set of news items can be generated in a scenario in which people talk about a restaurant in their social media feeds.

Example 2.2. Consider the NKB from Example 1.1 (Figure 1), and let us take user *Dan*'s perspective when the people he is connected to give their opinions and feedback on a certain restaurant, which we illustrate in Figure 2. These posts refer to certain aspects of the restaurant, which we

F	**Alice**: *It used to be really hard to get a table at #socialResto, but today I heard that they're taking reservations by phone booking and have a new bigger parking area.*	$(Alice, \epsilon, f)$ $(Alice, \beta, a)$
T	**Fred**: *Last Saturday I tried the vegetarian menu at #socialResto. I liked it. They had many dishes and the maitre was very attentive.*	$(Fred, \gamma, a)$ $(Fred, \zeta, a)$
W	**Beth**: *I saved time thanks to the new phone booking service at #socialResto. The dishes were exquisite and they had vegetarian options, too.*	$(Beth, \epsilon, a)$ $(Beth, \delta, a)$ $(Beth, \gamma, a)$
F	**Eric**: *Last night my wife and I had to wait about an hour to get a table at #socialResto. At least it was easy to find a parking space.*	$(Eric, \neg\epsilon, a)$ $(Eric, \beta, a)$
T	**Cate**: *Last year I went to #socialResto the waiter delayed in taking my order, but the food was great. It's been a while, I wonder if things have changed...*	$(Cate, \neg\zeta, r)$ $(Cate, \delta, r)$
T	**Alice**: *Two years ago I went to #socialResto and the prices were expensive and they had no veggie menu. I heard this has changed, but I am not sure.*	$(Alice, \neg\alpha, r)$ $(Alice, \neg\gamma, r)$
W	**Beth**: *I had to drive around for 15' until I found parking at #socialResto.*	$(Beth, \neg\beta, a)$
W	**Beth**: *My sister told me her bad experience with a waiter at #socialResto. My one previous experience was good, but now I don't know what to think about it.*	$(Beth, \zeta, r)$
F	**Cate**: *I just booked a table at #socialResto for a family lunch next Sunday. It was very easy.*	$(Cate, \epsilon, a)$
F	**Eric**: *Yesterday I was in the mood for a veggie dish at #socialResto but they said they only available on weekends. Also, the waiter was not very polite.*	$(Eric, \neg\gamma, a)$ $(Eric, \neg\zeta, a)$
T	**Fred**: *I used to drive to #socialResto but since I moved a block away from there I just walk. I never had problem with parking, but I heard that it's not so easy now.*	$(Fred, \beta, r)$

α: *suitable prices* β: *ample parking space* γ: *veggie friendly*
δ: *good food* ϵ: *table availability* ζ: *kind service*

Fig. 2. Posts seen by Dan in his feed. Social media post sources are Facebook, Twitter, and Weibo, represented by squares labeled **F**, **T**, and **W** on the left-hand side, respectively.

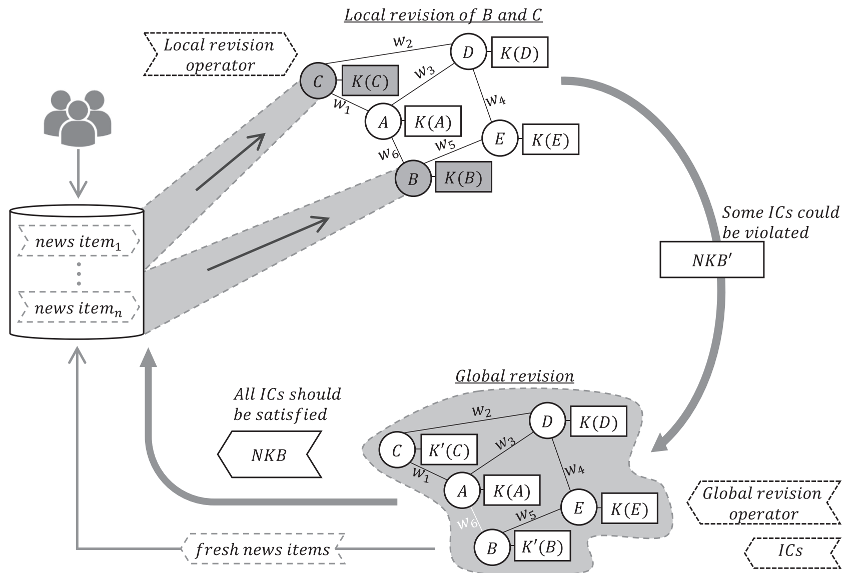

Fig. 3. Outline of the general belief revision process over a Network Knowledge Base.

will map to literals for convenience as follows: suitable prices (α), ample parking (β), vegetarian menu (γ), good food (δ), table availability (ϵ), and kind service (ζ). The figure shows the original posts on the left-hand side, and how they can be represented as news items on the right.

Summarizing, Dan sees the following news items in his feed (here, organized by literal); note that, for clarity, henceforth we use the vertices' labels instead of their names (so, for instance, "*Alice*" instead of "v_1"):

$$\begin{array}{rllll}
price: & (Alice, \neg\alpha, r), & & & \\
parking: & (Beth, \neg\beta, a), & (Alice, \beta, a), & (Eric, \beta, a), & (Fred, \beta, r), \\
veggie: & (Beth, \gamma, a), & (Fred, \gamma, a), & (Eric, \neg\gamma, a), & (Alice, \neg\gamma, r), \\
food: & (Cate, \delta, r), & (Beth, \delta, a), & & \\
table: & (Alice, \epsilon, a), & (Beth, \epsilon, a), & (Cate, \epsilon, a), & (Eric, \neg\epsilon, a), \\
service: & (Fred, \zeta, a), & (Beth, \zeta, r), & (Eric, \neg\zeta, a), & (Cate, \neg\zeta, r).
\end{array}$$

It is clear that there is considerable disagreement among his connections regarding several aspects: for instance, Cate and Beth on "good food" (δ) or Fred and Eric on "veggie friendly" (γ). Other disagreements may seem less problematic, such as Beth and Eric on "kind service" (ζ), since one adds the literal while the other removes its negation. ∎

With the basic definitions in place, we are ready to formulate and address belief dynamics in NKBs, which is the topic of the next section.

3 BELIEF DYNAMICS IN NETWORK KNOWLEDGE BASES

The overall belief dynamics process we propose is as follows: Consider Figure 3, where we have agents producing news items in one or more social platforms; depending on who follows whom, these news items will appear in different users' feeds (in the figure, we have only B and C receiving news items). A *local belief revision* process is then carried out over $K(B)$ and $K(C)$, taking into account these users' own perspective in the NKB. As a result of these operations, the new resulting network KB (NKB') might violate one or more integrity constraints, since each local revision was carried out in parallel—a *consistency/integrity maintenance* process is thus required. The output of the global process might make structural changes to the underlying network; for instance, agent B might no longer follow agent C on Instagram. This generates "fresh" news items, which take part in the next cycle.

In this article, we will only address the local revision step—global revision will be tackled in future work. A key aspect of the effort of developing local revision operators for the setting of social platform users is that user behaviors typically go beyond what is modeled by classical belief dynamics. Consider for instance a user who is simply in complete opposition to what some other user or group of users post; classical postulates do not capture this generality, and this is one of the differences between our work and he classical setting. Nevertheless, as we describe below, in this article we focus on the most basic kinds of operators, laying the groundwork for future developments.

Basic Assumptions. As described above, we assume that local revisions are carried out in parallel and synchronously, triggered by a global "clock." As a consequence of this, and the assumption that sets of news items are consistent, we assume that all sets of news items only contain the last decision regarding a given predicate symbol; e.g., if agent o removed α (news item (o, α, r)) and later added it again (news item (o, α, a)) before local revision is carried out, then the input will only contain the news item (o, α, a). Furthermore, we assume that each news item takes part in a single local revision process for each node—i.e., news items are "consumed" after each local revision.

In the rest of this section, we will present a set of rationality postulates that provide the basis for theoretical characterizations of different kinds of local revision operators, before moving on to their algorithmic characterization in Section 4.

3.1 Local NKB Belief Revision: Preliminary Definitions and Notation

In the following, we denote with $\mathcal{NKB}$ the universe of all possible network knowledge bases, and with $\mathcal{P}$ the universe of all possible news items. Given a Network Knowledge Base $\textsc{nkb} = (V, E, l_{vert}, l_{edge}, K)$, a consistent set of news items $P = \{p_1, \dots, p_n\}$, and vertex $v \in V$, we use $P_v = \{p = (o, l, d) \in P \mid (o, v) \in E\}$ to denote the set of *news items seen by* v. Furthermore, given news item $p = (o, l, d) \in \mathcal{P}$, we use $src(p) = o$, $lit(p) = l$, and $dec(p) = d$ to denote the *origin*, *literal*, and *decision* involved in p, respectively. Given $P \in \mathcal{P}$, $lit(P)$ is the set consisting of the literals of each $p \in P$; similarly, $lit_{neg}(P) = \{\neg l \mid l \in lit(P)\}$. Finally, for ease of presentation we assume that for literal l expressions of the form "$\neg\neg l$" are always simplified to "l".

We now provide a first definition of local NKB belief revision operator, which will then serve as the basis for more refined ones guided by the satisfaction of a subset of our postulates.

Definition 3.1. Local network knowledge base revision operators are partial functions of the form $\circledast : \mathcal{NKB} \times V \times 2^{\mathcal{P}} \rightarrow \mathcal{NKB}$, where the sets of news items are assumed to be consistent (cf. Definition 2.6).

Given a Network Knowledge Base $\textsc{nkb} \in \mathcal{NKB}$, a local revision operator $\circledast$, and a vertex $v \in V$, from now on we use the following notation to refer to the result of a local revision operation:

$$\textsc{nkb}' = \circledast(\textsc{nkb}, v, P) = (V', E', l'_{vert}, l'_{edge}, K').$$

Furthermore, given set of news items P, we use the following to denote the sets of all positive/added and negative/removed news items in P, respectively:

$$P^+ = \{p \in P \mid dec(p) = a\}$$
$$P^- = \{p \in P \mid dec(p) = r\}.$$

That is, P^{+} is the set of news items that refer to additions, while P^{-} is the set of those that refer to removals; note that it always holds that $P = P^{+} \cup P^{-}$.

We propose two sets of postulates as reasonable properties for local NKB revision: Those related only with epistemic issues and others that involve network properties. *Note that not all postulates are meant to hold for all operators—we will discuss this further in Section 4.*

3.2 Local NKB Belief Revision: Epistemic Postulates

Notation: $K'(v)$ denotes the revised local knowledge base, and P_v the set of news items seen by node v.

- **Inclusion:** $K'(v) \subseteq K(v) \cup lit(P_v) \ \cup \ lit_{neg}(P_v)$.

  No unwarranted information should be added as part of a revision.

- **Weak Positive Success:** $lit(P_v^{+}) \subseteq K'(v)$ when $K(v) \cup lit(P_v^{+})$ is consistent.

  Additions are guaranteed to be accepted if they are consistent with the KB.

- **Negative Success:** $lit(P_v^{-}) \cap K'(v) = \emptyset$.

  The literals in the input relating to removals should not be present in the revised KB.

- **Consistency:** $K'(v) \nvdash \bot$.

  The revised KB must be consistent.

- **Vacuity 1:** If $l \notin K(v)$ and $\neg l \notin K(v)$, then:

  — If $lit(p) \neq l$ and $lit(p) \neq \neg l$ for all $p \in P_v$, then $l \notin K'(v)$.
  — Otherwise, if $lit(p) = l$ implies $dec(p) = r$, for all $p \in P_v$, then $l \notin K'(v)$.

If there is no evidence in the input about adding a literal, and the local KB does not contain the literal, then it should not be added as part of the revision. In other words, no information is added to the local KB without justification.

- **Vacuity 2:** If $l \in K(v)$, then:

  — If $lit(p) \neq l$ and $lit(p) \neq \neg l$ for all $p \in P_v$, then $l \in K'(v)$.
  — Otherwise, if either $lit(p) = \neg l$ implies $dec(p) = r$, or $lit(p) = l$ implies $dec(p) = a$, for all $p \in P_v$, then $l \in K'(v)$.

As a dual of Vacuity 1, this postulate identifies two conditions under which a literal that is part of the KB should be kept as part of the revision. The first is whenever there is no reference to the literal in the set of news items. The second is divided into two cases: (i) whenever all references to the negation of the literal have remove as a decision or (ii) whenever all references to the literal itself have add as a decision.

- **Weak Vacuity 1:** If $l \notin K(v)$, then, if $lit(p) \neq l$ and $lit(p) \neq \neg l$ for all $p \in P_v$, then $l \notin K'(v)$.

  An element is not added into the local KB for which no information is received.

- **Weak Vacuity 2:** If $l \in K(v)$, then, if $lit(p) \neq l$ and $lit(p) \neq \neg l$ for all $p \in P_v$, then $l \in K'(v)$.

  No element is removed from the local KB without receiving information about its negation.

- **Congruence:** Given set of news items P''_v and the resulting revision $\textsc{nkb}'' = \circledast(\textsc{nkb}, v, P''_v) = (V'', E'', l''_{vert}, l''_{edge}, K'')$, if $P_v = P''_v$ then $K'(v) = K''(v)$.

If two sets of news items are equivalent,[3] then the respective local KBs should be identical—in other words, the result does not depend on the syntax used to express the input. In belief change, congruence conditions require that the result of changes (revisions or contraction) should not depend on syntactic properties of the sentences to be modified: only their logical content should count. These postulates in the AGM (Alchourrón, Gärdenfors, and Makinson) model [1] are also called *extensionality*. A logic is extensional if it allows logically equivalent sentences to be freely substituted for each other [33].

- **Removal Monotonicity:** Given set of news items P''_v and the resulting revision $\textsc{nkb}'' = \circledast(\textsc{nkb}, v, P''_v) = (V'', E'', l''_{vert}, l''_{edge}, K'')$, if $P^+_v = P''^+_v$ and $P^-_v \subseteq P''^-_v$, then $K''(v) \subseteq K'(v)$.

This postulate can be seen as a weaker version of the previous property, stating that if both inputs are identical with respect to the content they add, but one of the inputs removes a superset of what the other removes, then the output for the former is a subset of the output for the latter.

- **Uniformity:** If P and Q are two sets of news items such that $P = Q$, then $\circledast(\textsc{nkb}, v, P) = \circledast(\textsc{nkb}, v, Q)$.

This postulate is an adaptation of the classical Uniformity postulate from Reference [31], which was proposed for revisions where the epistemic input is a single sentence. Uniformity stipulates that if the literals in two sets of news items are consistent with the same subsets of the original local KB $K(v)$, then the respective erased sentences of $K(v)$ should be identical. In our setting, this postulate holds trivially, since we only consider literals.

3.3 Local NKB Belief Revision: Network Postulates

The following postulates describe how network properties affect how the local revision operation is carried out. Recall once again that $K'(v)$ is the revised local knowledge base and P_v is the set of news items seen by node v.

- **Local Effect:** $\forall w \in V$ s.t. $w \neq v, K'(w) = K(w)$.

Applying a local revision operator must not have any effect on other agents' KBs.

- **Structural Preservation:** $V = V', E = E', l_{vert} = l'_{vert}$, and $l_{edge} = l'_{edge}$.

The set of vertices, edges, and labeling functions remain unchanged after a local revision operation.

The next two postulates make use of two values, *wPos* and *wNeg*, calculated as follows:

$$wPos = wf^+(\textsc{nkb}, v, P, l) \text{ and}$$
$$wNeg = wf^-(\textsc{nkb}, v, P, l).$$

where wf^+ and wf^- are real functions ranging over $[0, 1]$. Intuitively, these functions represent (weighted) votes in favor or against a given literal in a set of news items. We only impose that the functions range over the real unit interval; if other properties—like monotonicity—are desired, then they can be designed to satisfy them. Two simple examples of such functions are: (i) direct counting, where for each literal l function wf^+ counts how many times l was added, while wf^- counts how many times $\neg l$ was added; and (ii) a variant of the previous function in which news itsms for specific users are ignored (or counted as negative when they are in fact positive).

- **Weak Voting:** Let $l \in Pred$ such that $l \in lit(P)$ or $\neg l \in lit(P)$. Then, if $wPos > wNeg$, then $\neg l \notin K'(v)$; if $wPos < wNeg$, then $l \notin K'(v)$.

[3] Note that here we use "=" instead of "≡" because we only deal with literals.

This postulate is similar to the Majority postulate proposed in Reference [39]; the basic idea is to allow votes to be weighted according to the relationship between the agent and the origin of each news item. The intuition behind $wf^+$and $wf^-$is that both functions calculate a value representing opinions for and against a certain literal, respectively, taking into account network properties.

- **Strong Voting:** Let $l \in Pred$ such that $l \in lit(P)$ or $\neg l \in lit(P)$. Then, if $wPos > wNeg$, then $l \in K'(v)$; if $wPos < wNeg$, then $\neg l \in K'(v)$.

As a variant of Weak Voting, when the reasons for a literal outweigh those against it, the literal should be included in the resulting KB; otherwise, its negation should be included.

There are some basic relations among our postulates. Trivially, *Vacuity 1* and *Vacuity 2* logically imply *Weak Vacuity 1* and *Weak Vacuity 2.* Finally, we have the following less direct result.

Proposition 3.1. *If a local NKB revision operator satisfies Inclusion, then it also satisfies Weak Vacuity 1.*

Proof. Let NKB $= (V, E, l_{vert}, l_{edge}, K)$, $v \in V$, P_v be a set of a news items for v, and $l \notin K(v)$. We want to show that if $lit(p) \neq l$ and $lit(p) \neq \neg l$ for all $p \in P_v$, then $l \notin K'(v)$.

Let us suppose toward a contradiction that the local revision operator satisfies *Inclusion* but it does not satisfy *Weak Vacuity 1*, i.e., there is l such that for all $p \in P_v$ we have $lit(p) \neq l$ and $lit(p) \neq \neg l$, and $l \in K'(v)$. By *Inclusion* we know that $l \in K(v) \cup lit(P_v) \cup lit_{neg}(P_v)$; thus, $l \in K(v)$, $l \in lit(P_v)$, or $l \in lit_{neg}(P_v)$. Next, by hypothesis we have that $l \notin K(v)$, so it must be the case that $l \in lit(P_v)$ or $l \in lit_{neg}(P_v)$, which contradicts our hypothesis. □

The most important difference between our approach and the classical postulates from the belief revision literature is that—even though we restrict the language to literals—in our case the epistemic input is more complex, since it contains additional information about each individual literal (i.e., its origin, and whether the decision was to add or remove it). Therefore, the decision of what to accept from the input and/or how to make changes to the KB in response to its contents may also depend on other aspects at a more global level, such as the relationships between users or their individual features.

The following example illustrates the application of some of the previous postulates.

Example 3.1. Consider the set of news items from Example 2.2 and the scenario from the running example; let us see some of the options that Dan has in this case. Recall that $K(Dan) = \{\neg\beta, \neg\delta, \eta\}$ and the news item set $P_{Dan} = \{ (Alice, \neg\alpha, r),\ (Beth, \neg\beta, a),\ (Alice, \beta, a),\ (Eric, \beta, a),\ (Fred, \beta, r),$ $(Beth, \gamma, a),\ (Fred, \gamma, a),\ (Eric, \neg\gamma, a),\ (Alice, \neg\gamma, r),\ (Cate, \delta, r),\ (Beth, \delta, a),\ (Alice, \epsilon, a),$ $(Beth, \epsilon, a),\ (Cate, \epsilon, a),\ (Eric, \neg\epsilon, a),\ (Fred, \zeta, a),\ (Beth, \zeta, r),\ (Eric, \neg\zeta, a),\ (Cate, \neg\zeta, r) \}$:

- For the analysis of both Success postulates, we first need the definition of the next two sets:

$$P^+_{Dan} = \Big\{ (Beth, \neg\beta, a),\ (Alice, \beta, a), (Eric, \beta, a),\ (Beth, \gamma, a),\ (Fred, \gamma, a),$$
$$(Eric, \neg\gamma, a),\ (Fred, \zeta, a),\ (Beth, \delta, a), (Alice, \epsilon, a),\ (Beth, \epsilon, a),$$
$$(Cate, \epsilon, a),\ (Eric, \neg\epsilon, a), (Eric, \neg\zeta, a) \Big\}, \textit{and}$$
$$P^-_{Dan} = \Big\{ (Alice, \neg\alpha, r),\ (Fred, \beta, r),\ (Cate, \delta, r),\ (Alice, \neg\gamma, r), (Beth, \zeta, r),$$
$$(Cate, \neg\zeta, r)\Big\}.$$

Regarding *Weak Positive Success*, the postulate's premise does not hold, since $K(Dan) \cup lit(P^+_{Dan}) = \{\neg\beta, \neg\delta, \eta\} \cup \{\beta, \neg\beta, \gamma, \neg\gamma, \zeta, \neg\zeta, \delta, \epsilon, \neg\epsilon\}$ is clearly inconsistent; thus, this postulate is satisfied even if none of the symbols from the input are added to the KB. However, no literal from $lit(P_{Dan^-}) = \{\neg\alpha, \beta, \delta, \neg\gamma, \zeta, \neg\zeta\}$ can be included in $K(Dan)$ as the result of a revision satisfying *Negative Success*.

- For Vacuity 1 to hold, he should not add completely new knowledge into his KB, such as "the restaurant has delivery service" represented by θ. Moreover, he has no reason to believe that the restaurant does not have suitable prices—this is caused by the presence of news item $(Alice, \neg\alpha, r)$. In other words, he should not add $\neg\alpha$ to his KB. However, he could include γ, $\neg\gamma$, ϵ, $\neg\epsilon$, ζ, $\neg\zeta$, or α into his KB (but not $\neg\alpha$). Now, considering $\neg\beta$ and $\neg\delta$ that belong to $K(Dan)$, *Vacuity 1* holds trivially, since the postulate's precondition does not hold for those literals.
- Considering *Vacuity 2*, because not all news items referring to the negation of $\neg\delta$ (i.e., δ) remove it (e.g., $(Beth, \delta, a)$), and none of the news item contains $\neg\delta$, the postulate holds regardless of whether Dan keeps or eliminates $\neg\delta$.
  Regarding $\neg\beta$, for *Vacuity 2* to hold Dan should not remove that literal from his KB, since the (ii) of the postulate's second condition holds; i.e., every news item containing the literal $\neg\beta$ has *addition* as decision (the only news item containing the literal is $(Beth, \neg\beta, a)$). However, insomuch as the postulate's premise does not hold for the symbol α (i.e., neither α nor $\neg\alpha$ belong to the KB), Dan can ignore or add either α or $\neg\alpha$ to his KB. Finally, because no new information respect to η is in the feeds, Dan cannot remove it from his KB if he wants to satisfy *Vacuity 2*.
- Now, to satisfy Weak Vacuity 1, Dan should not add unjustified knowledge into his KB, as with the Vacuity 1 postulate. However, he could add γ, $\neg\gamma$, ϵ, $\neg\epsilon$, ζ, or $\neg\zeta$ into his KB. In contrast to the Vacuity 1 postulate, since at least one news item refers to α or $\neg\alpha$, he can add either α or $\neg\alpha$.
- Moreover, since Weak Vacuity 2 does not hold for $\neg\beta$ nor $\neg\delta$, Dan could remove them. Lastly, similar to the Vacuity 2 case, he cannot remove η from his KB as a result of this postulate.
- Finally, let us suppose we want to satisfy both Weak and Strong Voting in the revision of $K(Dan)$ by P_{Dan}. Let us analyze the cases of $\neg\beta$, $\neg\delta$, and ϵ, and further suppose that Dan's valuation functions are as follows:

$$wPos = wf^+\left(NKB, Dan, P_{Dan}, l\right) = \frac{\sum_{e \in I}\left(wgh(e) \cdot w^+\right)}{|I|},$$

$$wNeg = wf^-\left(NKB, Dan, P_{Dan}, l\right) = \frac{\sum_{e \in J}\left(wgh(e) \cdot w^-\right)}{|J|},$$

  where
  - given $\langle b, w\rangle \in l_{edge}$, $wgh(\langle b, w\rangle) = w$; this function returns the weight of a given edge's label. Furthermore, we use an additional function for defining the sets I and J: given two vertices $v_1, v_2 \in V$, $highestWeight(v_1, v_2) = \langle b, w\rangle$, such that w is the highest weight among all the edge's labels between these two vertices.
  - $I = \Big\{\langle b, w\rangle \in l_{edge}(e) \mid e = (Dan, src(p)), w = highestWeight(e),$
    $\Big(lit(p) = l \text{ y } dec(p) = a\Big) \text{ or } \Big(lit(p) = \neg l \text{ y } dec(p) = r\Big), \text{con } p \in P_{Dan}\Big\}.$

    Since there can be more than one label in an edge between two vertices, I is the set of the edges with the highest weight among Dan and each one of those connections who either adopt a certain literal or remove its negation.

- $J = \Big\{\langle b, w\rangle \in l_{edge}(e) \mid e = (Dan, src(p)), w = highestWeight(e),$
  $\Big(lit(p) = \neg l$ y $dec(p) = a\Big)$ or $\Big(lit(p) = l$ y $dec(p) = r\Big)$, con $p \in P_{Dan}\Big\}$.

  The set J is similar to I, but in this case the connections who either remove a certain literal or adopt its negation are considered.
- $w^+, w^- \in [0, 1]$ are the *weakening values*. They represent the agent's tendency to give more importance to positive or negative opinions about a certain literal.

Let us suppose that Dan gives more importance to positive information by defining $w^+ = 1$ and $w^- = 0.5$ for *wPos* and *wNeg*, respectively.

- Considering $\neg\beta$:

  $I_{\neg\beta} = \{$ $(Dan, Beth)$ (max label is *work partner*, weight 0.9),
  $(Dan, Fred)$ (max label is *fan*, weight 0.6)$\}$

  $J_{\neg\beta} = \{$ $(Dan, Alice)$ (max label is *couple*, weight 0.8),
  $(Dan, Eric)$ (max label is *student*, weight 0.6)$\}$.

  For ease of presentation, in this example sets I and J include information on the pair (v_1, v_2) for each label so it is possible to note the edges that they are attached to. Then, $wPos = (0.9 \times 1 + 0.6 \times 1)/2 = 0.75$, and $wNeg = (0.8 \times 0.5 + 0.6 \times 0.5)/2 = 0.175$.
  To satisfy Weak Voting, Dan must not adopt β but he can remove $\neg\beta$. However, to satisfy Strong Voting he cannot remove $\neg\beta$ from his KB.
- Now, considering $\neg\delta$ we have the following:

  $I_{\neg\delta} = \{$ $(Dan, Cate)$ (max label is *employee*, weight 0.4)$\}$

  $J_{\neg\delta} = \{$ $(Dan, Beth)$ (max label is *work partner*, weight 0.9)$\}$.

  So, $wPos = (0.4 \times 1)/1 = 0.4$, and $wNeg = (0.9 \times 0.5)/1 = 0.45$.
  To satisfy Weak Voting, Dan should remove $\neg\delta$ from his KB and—if he wants, he can adopt δ. On the contrary, to satisfy Strong Voting he must add δ.
- Regarding ϵ, we have:

  $I_\epsilon = \{$ $(Dan, Alice)$ (max label is *couple*, weight 0.8),
  $(Dan, Beth)$ (max label is *work partner*, weight 0.9),
  $(Dan, Cate)$ (max label is *employee*, weight 0.4)$\}$,

  $J_\epsilon = \{$ $(Dan, Eric)$ (max label is *student*, weight 0.6)$\}$.

  Then, $wPos = (0.8 \times 1 + 0.9 \times 1 + 0.4 \times 1)/3 = 0.7$, and $wNeg = (0.6 \times 0.5)/1 = 0.3$.
  Dan cannot add $\neg\epsilon$ into his KB if he wants to satisfy Weak Voting. On the contrary, to satisfy Strong Voting Dan must adopt ϵ.
- Regarding α, γ, and ζ, we have the following:

  $I_\alpha = \{$ $(Dan, Alice)$ (max label is *couple*, weight 0.8)$\}$

  $J_\alpha = \emptyset$

  Then, $wPos = (0.8 \times 1)/1 = 0.8$, and $wNeg = 0$.

  $I_\gamma = \{$ $(Dan, Alice)$ (max label is *couple*, weight 0.8),
  $(Dan, Beth)$ (max label is *work partner*, weight 0.9),
  $(Dan, Fred)$ (max label is *fan*, weight 0.6)$\}$,

  $J_\gamma = \{$ $(Dan, Eric)$ (max label is *student*, weight 0.6)$\}$.

  So, $wPos = (0.8 \times 1 + 0.9 \times 1 + 0.6 \times 1)/3 \approx 0.77$, and $wNeg = (0.6 \times 0.5)/1 = 0.3$.

$I_\zeta = \{$ (Dan, $Cate$) (max label is *employee*, weight 0.4),
(Dan, $Fred$) (max label is *fan*, weight 0.6)$\}$

$J_\zeta = \{$ (Dan, $Beth$) (max label is *work partner*, weight 0.9),
(Dan, $Eric$) (max label is *student*, weight 0.6)$\}$.

Next, $wPos = (0.4 \times 1 + 0.6 \times 1)/2 = 0.5$, and $wNeg = (0.9 \times 0.5 + 0.6 \times 0.5)/2 = 0.375$. According to these results, because of *Weak Voting* Dan cannot add $\neg\alpha$, $\neg\gamma$, nor $\neg\zeta$ to his KB; however, he must add α, γ, and ζ to satisfy *Strong Voting*.

- With respect to η, Dan can keep or remove it. Unlike the previous postulates in this example, Dan can add unjustified information—e.g., θ: "the restaurant has delivery service"—and yet still satisfy both Voting postulates. ■

The following definition characterizes the operators considered to be minimally reasonable.

Definition 3.2. A local NKB revision operator $\circledast$ is *basic* if it satisfies Structural Preservation, Local Effect, Consistency, Uniformity, and Inclusion.

Using this basic set of postulates, we will define three families of operators. The first two, called *restrained* and *weakly restrained*, model revisions that are closer to rational reactions to the content of news items; the other, called *social*, is meant to capture revisions that are based on other kinds of influences such as the opinions expressed by others on social media. In this article, we will focus on these three families to build a foundation that can later be extended to other kinds of operators; going beyond this initial step is outside the scope of this article, since the range of possible user behaviors is too great. Future work will be dedicated to the development of operators that model specific kinds of online users; as discussed in the related work section, we have already taken some steps in this direction.

Definition 3.3. Let $\circledast$ be a basic local NKB revision operator:

- $\circledast$ is *restrained* if it satisfies Congruence, Vacuity 1, and Vacuity 2.
- $\circledast$ is *weakly restrained* if it satisfies Removal Monotonicity, Weak Vacuity 1, and Weak Vacuity 2.
- $\circledast$ is *social* if it satisfies Weak Vacuity 1, Weak Vacuity 2, and either Weak Voting or Strong Voting.

In the following example, we illustrate some of the differences among these general classes of operators.

Example 3.2. In Example 3.1 we analyzed the restrictions enforced by some postulates in isolation. Here, we continue that example and compare how the operators make the possible changes with respect to the symbols implicated in the local revisions of $K(Dan)$ by P_{Dan} (recall that simple weighted counts are used for the wf^+ and wf^- functions):

- Symbol α ($\alpha \notin K(Dan)$, $\neg\alpha \notin K(Dan)$)
  - Restrained operator: Dan cannot add $\neg\alpha$ but he can add α.
  - Weakly Restrained operator: He can add either α or $\neg\alpha$.
  - Social operator: He cannot add $\neg\alpha$, and he must add α into his KB.
- Symbol β ($\neg\beta \in K(Dan)$)
  - Restrained and Weakly Restrained operator: Dan can keep or remove $\neg\beta$.
  - Social operator: For Weak Voting he can remove $\neg\beta$ but cannot add β. However, if Strong Voting is used, then Dan cannot remove $\neg\beta$. Recall that the social operator requires that one of the two Voting postulates be satisfied.

<table>
<tr><th rowspan="2">Symbol s</th><th rowspan="2">$s \in lit(P_{Dan})$?
$s \in K(Dan)$?</th><th colspan="3">Operator</th></tr>
<tr><th>Restrained</th><th>Weakly Restrained</th><th>Social</th></tr>
<tr><td>α</td><td>Yes, No</td><td>α can be added</td><td>α or $\neg\alpha$ can be added</td><td>α must be added</td></tr>
<tr><td rowspan="2">β</td><td rowspan="2">Yes, Yes</td><td rowspan="2">$\neg\beta$ cannot be removed</td><td rowspan="2">$\neg\beta$ can be kept or removed</td><td>(WV) $\neg\beta$ can be removed, β cannot be added</td></tr>
<tr><td>(SV) $\neg\beta$ cannot be removed</td></tr>
<tr><td>γ, ϵ, ζ</td><td>Yes, No</td><td colspan="2">γ or $\neg\gamma$, ϵ or $\neg\epsilon$, and ζ or $\neg\zeta$ can be kept, added, or ignored</td><td>γ, ϵ, and ζ must be added</td></tr>
<tr><td rowspan="2">δ</td><td rowspan="2">Yes, Yes</td><td colspan="2" rowspan="2">$\neg\delta$ can be kept or removed</td><td>(WV) $\neg\delta$ can be removed, δ can be added</td></tr>
<tr><td>(SV) δ must be added</td></tr>
<tr><td>η</td><td>No, Yes</td><td colspan="3">η cannot be removed</td></tr>
<tr><td>θ</td><td>No, No</td><td colspan="3">New information cannot be "made-up"</td></tr>
</table>

Fig. 4. Summary of all possible combinations according to each local revision operator per symbol from Example 3.2. "WV" and "SV" stand for Weak Voting and Strong Voting, respectively.

- Symbols $\gamma, \epsilon,$ and ζ (none of these symbols belongs to $K(Dan)$):
  - Restrained and Weakly Restrained operators: Dan can add γ, $\neg\gamma$, ϵ, $\neg\epsilon$, ζ, or $\neg\zeta$ into his KB, or do nothing.
  - Social operator: He cannot add $\neg\zeta$, $\neg\gamma$, nor $\neg\epsilon$; he must add ζ, γ, and ϵ into his KB.
- Symbol δ ($\neg\delta \in K(Dan)$):
  - Restrained and Weakly Restrained operators: Dan can keep or remove $\neg\delta$.
  - Social operator: For Weak Voting, Dan has to remove $\neg\delta$ and—if he wants—he can add δ; for String Voting he must add δ into his KB.
- Concerning $\eta \in K(Dan)$: Regardless of the operator, Dan cannot remove η from his KB. Although we saw in the previous example that both Voting postulates allow to change η freely, social operators further require Weak Vacuity 2, which precludes the removal.
- Regarding the possibility of "making up" new knowledge, the three operators avoid it, since they must satisfy *Inclusion*.

  To illustrate the variety of possible operator behaviors in this example, the table in Figure 4 summarizes all possible operations for each symbol. ■

In the following section, we continue with the algorithmic characterization of the first two families of operators.

4 A CONSTRUCTION FOR *RESTRAINED* AND *WEAKLY RESTRAINED* OPERATORS

In Section 3, we provided the basis for the theoretical characterization of local NKB revision operations—essentially, such a characterization is based on the satisfaction of a specific set of properties that the revision must satisfy. In this section, we take the next step toward implementing specific operators by devising an *algorithmic* characterization for *restrained* and *weakly restrained* operators (cf. Definition 3.3). We ultimately conclude that the two characterizations are closely related.

We begin by analyzing different kinds of relations that exist between an agent's beliefs and a news item, or between two news items; as we discuss below, this will define one of the main building blocks of our operator.

Definition 4.1 (Relations Between Local KBs and News Items). Let *KB* be a local knowledge base, and p be a news item. There are four types of relations that can hold between a knowledge base and a news item:

- Type 1 (T1 relation, or "*hard conflict*"): $\neg lit(p) \in KB$ and $dec(p) = a$.
- Type 2 (T2 relation, or "*soft conflict*"): $lit(p) \in KB$ and $dec(p) = r$.

- Type 3 (T3 relation, or "*negative suggestion*"): $lit(p) \notin KB$ and $\neg lit(p) \notin KB$, and $dec(p) = r$.
- Type 4 (T4 relation, or "*positive suggestion*"): $lit(p) \notin KB$ and $\neg lit(p) \notin KB$, and $dec(p) = a$.

Intuitively, hard conflicts (T1) occur when the contents of the KB and the news item are in direct conflict with each other; a decision must be made so that only one of the two is kept. In soft conflicts (T2), the news item can be seen as an undercut for the agent's beliefs; a decision may or may not be made between the two pieces of information as a response. Finally, in negative and positive suggestions (T3 and T4), a decision may or may not be made regarding content that is not present in the knowledge base. We now define the corresponding relations between two news items.

Definition 4.2 (Relations Between News Items). Let p_1, p_2 be news items. There are two types of relations that can hold between any such pair:

- Type 1 (T1 relation, or "*hard conflict*"): $dec(p_1) = a$, $dec(p_2) = a$, and $lit(p_1) = \neg lit(p_2)$.
- Type 2 (T2 relation, or "*soft conflict*"): $dec(p_1) = a$, $dec(p_2) = r$, and $lit(p_1) = lit(p_2)$.

These relations are then used in the definition of the *local revision choice graph.*

Definition 4.3. Let KB be a local knowledge base and P be a set of news items. The *revision choice graph* is defined as an undirected graph $G = (V, E)$ such that:

- *Nodes:* There is one node in V for each news item $p \in P$, one node for each s such that either $s \in lit(p)$ or $\neg s \in lit(p)$, for some $p \in P$, one node for each element in KB, and one additional node labeled *inertia*.
- *Edges:* There is one edge in E between each pair of nodes such that:
  - the corresponding literal in KB and the corresponding news item are in one of the relations defined in Definition 4.1;
  - the corresponding news items are in one of the relations defined in Definition 4.2; and
  - between *inertia* and all nodes corresponding to news items p such that $lit(p) \notin KB$, $\neg lit(p) \notin KB$, and either
    - $dec(p) = a$, or
    - $dec(p) = r$ whenever there is no news item p' such that $lit(p) = lit(p')$ or $lit(p) = \neg lit(p')$, and $dec(p') = a$.

Essentially, choice graphs show us the pairs of nodes for which a decision should be made; the special node *inertia* represents the predicate symbols for which there is no position in the knowledge base (i.e., neither l nor $\neg l$ belong to the KB). Figure 5 shows the choice graph for the running example. Regarding the negative suggestions (T_3), note that every time a literal is only referenced with a *remove* decision in the news item set (such as $\neg\alpha$ in the running example), the conflict is represented between the remove node and *inertia*.

As we discuss next, the choice graph is the basic semantic structure on which we build our algorithmic characterization. Essentially, the unordered binary relations contained in the graph indicate a choice that will ultimately decide the contents of the knowledge base after the revision. This initial formulation of the graph is general and, depending on the properties that we wish to build into the operator, refinements need to be made by changing undirected edges to directed ones, signaling that the operator is constrained to make a specific choice for that pair—this is the same kind of approach as taken in, for instance, defining preferred repairs in inconsistent databases [58]. In this article, we will focus only on the *restrained* and *weakly restrained* operator types defined above.

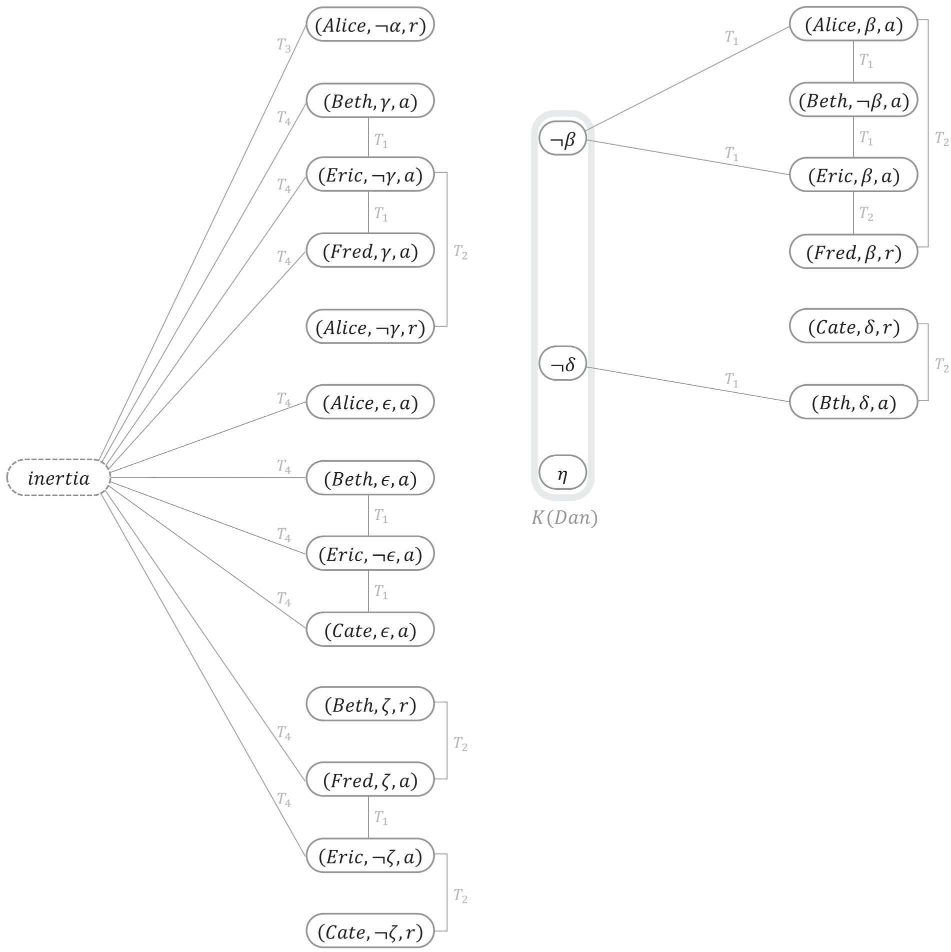

Fig. 5. Revision choice graph generated from the set of news items from Example 2.2. Since $K(Dan) = \{\neg\beta, \neg\delta, \eta\}$, there is an edge between each node representing a symbol that does not belong to the KB (i.e., α, γ, ϵ, and ζ) and the *inertia* node.

4.1 Data Structures

Starting from the choice graph introduced above, we define several data structures that will play key roles in our main algorithm; we describe each of them in turn before presenting their use. We begin with summarized predicate graphs, an example of which is presented in Figure 6. Such graphs arise from a set of patterns (shown in Figure 7), and a record data structure is associated with their nodes (with fields as described in Figure 8).

4.1.1 Summarized Predicate Symbol Graphs. One of the main advantages of limiting the language in local knowledge bases to literals is that they can essentially be independently modeled and treated. We can thus take the general choice graph from Definition 4.3 and derive one *summarized predicate symbol graph* for each symbol s as follows; we assume that v is the node in the NKB whose knowledge base is being revised, and that we are given a set P of news items:

- If $s \in KB$, then the graph contains a node labeled "s"; if $\neg s \in KB$, then the graph contains a node labeled "$\neg s$". Otherwise, this node is labeled "$s_{inertia}$".

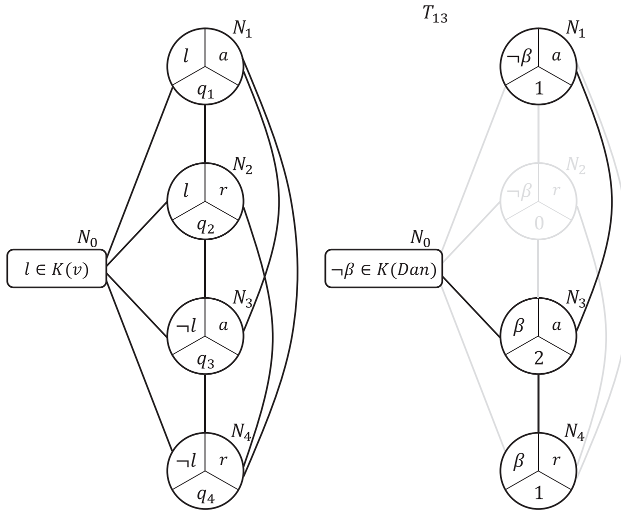

Fig. 6. Example summarized predicate symbol graphs: (Left) General data structure; (right) example instance (T_{13} from Figure 7), where nodes and edges that are not used are shown in gray. Note that, since the literal in question is $\neg\beta$, the literals in nodes N_1–N_4 are negated; in particular, since $\neg\neg\beta \equiv \beta$, nodes N_3 and N_4 are simply labeled "β".

- The graph contains one node for each of the four possible combinations of elements $p \in P$ such that $lit(p) = s$ or $lit(p) = \neg s$:
  (1) $lit(p) = s$ and $dec(p) = a$;
  (2) $lit(p) = s$ and $dec(p) = r$;
  (3) $lit(p) = \neg s$ and $dec(p) = a$; or
  (4) $lit(p) = \neg s$ and $dec(p) = r$.
  Each such node represents the set of elements in $p \in P$ satisfying those conditions, and are labeled with s, $dec(p)$, and the number of news items in P satisfying them.
- Node $s_{inertia}$ appears only when there are news items $p \in P$ such that $lit(p) = s$ or $lit(p) = \neg s$, $s \notin KB$, and $\neg s \notin KB$.
- Nodes corresponding to news items have an associated record (called *stateRecord*) of the form shown in Figure 8, holding the set of agents that posted that particular kind of news item—for computational convenience, it also stores the cardinality of these sets.
- Finally, the graph contains one edge between the nodes that represent nodes in the choice graph that have edges between them.

Figure 6 shows an example of a summarized predicate symbol graph where, for simplicity of presentation, we only display the number of agents and not the set associated with news item nodes. Note that the inertia nodes only appear when no removals are being proposed; this is to avoid redundant cases, since otherwise equivalent outcomes would arise—in these cases, *the removal nodes represent the decision to leave the knowledge base as it is with respect to that symbol.* Note that if the literal in question is already negated, then the literals in nodes N_3 and N_4 are simplified, since, as we mentioned above, $\neg\neg l \equiv l$.

These graphs are intuitively abstractions of the choice graph—as such, a key property is that there is a fixed number of possible instances of them that can arise in practice. In Figure 7, we provide the full list, giving each of them an identifier that will be used later. Note that whenever the literal in the KB is already negated, the literals in nodes N_3 and N_4 are simplified (i.e., without the negation), since $\neg\neg l \equiv l$, as was previously mentioned. For this reason, the conflict pattern for

#collapsed NIs	Input Pattern				ID
	numPosAdd	*numPosRem*	*numNegAdd*	*numNegRem*	
1	$\neq 0$	0	0	0	T_1
	0	$\neq 0$	0	0	T_2
	0	0	$\neq 0$	0	T_3
	0	0	0	$\neq 0$	T_4
2	$\neq 0$	$\neq 0$	0	0	T_5
	$\neq 0$	0	$\neq 0$	0	T_6
	$\neq 0$	0	0	$\neq 0$	T_7
	0	$\neq 0$	$\neq 0$	0	T_8
	0	$\neq 0$	0	$\neq 0$	T_9
	0	0	$\neq 0$	$\neq 0$	T_{10}
3	$\neq 0$	$\neq 0$	$\neq 0$	0	T_{11}
	$\neq 0$	$\neq 0$	0	$\neq 0$	T_{12}
	$\neq 0$	0	$\neq 0$	$\neq 0$	T_{13}
	0	$\neq 0$	$\neq 0$	$\neq 0$	T_{14}
4	$\neq 0$	$\neq 0$	$\neq 0$	$\neq 0$	T_{15}

Fig. 7. All possible combinations of summarized predicate symbol graphs.

Field	Content given symbol s and set of news items P
whoPosAdd	$\{o \mid (o, s, a) \in P_v\}$
whoPosRem	$\{o \mid (o, s, r) \in P_v\}$
whoNegAdd	$\{o \mid (o, \neg s, a) \in P_v\}$
whoNegRem	$\{o \mid (o, \neg s, r) \in P_v\}$
numPosAdd	$\mid whoPosAdd \mid$
numPosRem	$\mid whoPosRem \mid$
numNegAdd	$\mid whoNegAdd \mid$
numNegRem	$\mid whoNegRem \mid$

Fig. 8. Records associated with news item nodes in summarized predicate symbol graphs.

β is T_{13} instead of T_{11}; in other words, as *#collapsed NIs* (number of collapsed news items) is 3 (cf. Figure 7), the amounts for determining its pattern are as follows:

$$\begin{array}{lcll} numPosAdd & = & 1 & (\{(Beth, \neg\beta, a)\}), \\ numPosRem & = & 0, & \\ numNegAdd & = & 2 & (\{(Alice, \beta, a), (Eric, \beta, a)\}), \\ numNegRem & = & 1 & (\{(Fred, \beta, r)\}). \end{array}$$

A similar situation occurs with δ, whose conflict pattern is T_{10} instead of T_5. In other words, this situation must be considered whenever a symbol is in the KB and the literal is negated.

4.1.2 Strict Partial Orders. Finally, the structures discussed above are mapped to *possible outcomes* for the revised knowledge base. Taking advantage of the fact that summarized predicate symbol graphs can be enumerated, we analyzed all possible *directed* graphs arising from each undirected one (Types T_1–T_{15} in Figure 7)—the set of undominated nodes (or *skyline*) in each case denotes how the different pieces of information can be favored, and thus denote a possible way to revise the knowledge base given the set of news items seen by the agent. Figure 9 shows examples of such **strict partial orders (SPOs)** and their corresponding skylines; note that the one on the left-hand side is an instance of the general structure shown in Figure 6 (right).

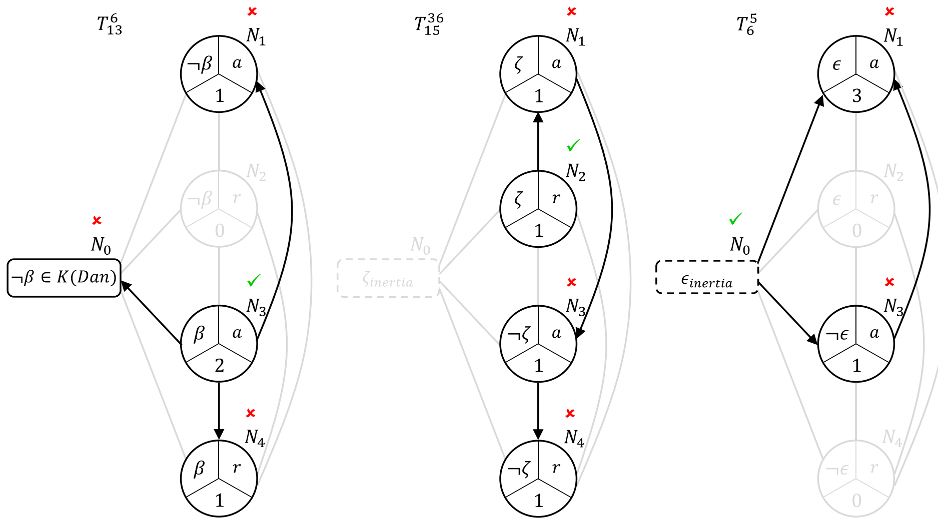

Fig. 9. Example strict partial orders; these instances correspond to T_{13}^{6}, T_{15}^{36}, and T_{6}^{5} from Figures 13–15. Dominated nodes are marked with a cross, while undominated ones (those belonging to the skyline) are marked with a check.

Going through all such possibilities and keeping only the ones that satisfy the postulates required by (*weakly*) *restrained* operator types,[4] we arrive at the list shown in Appendix A (Figures 13–15); we give each possibility an identifier that extends the ones from Figure 7. Figure 11 shows the array structure used in our algorithm to represent the set of SPOs thus computed.

4.2 The Main Algorithm

The pseudocode for Algorithm *reviseForRestrained* is presented in Figure 10; at this point, we have described in detail the building blocks used to derive the possible outcome KBs for the revision operation. In line 2, we compute the data structure shown in Figure 11; to represent the set of SPOs; then, for each predicate symbol we first determine the *conflict pattern*—i.e., the presence or absence of each possible type of news item—and then go on to determine its status with respect to the input knowledge base. Once this is established, we can find the relevant SPOs in the tables shown in Appendix A (Figures 13–15), which also depend on the user type (which can be either *restrained* or *weakly restrained*). This step makes a choice for each symbol, which depending on the status of that symbol in the KB, is always one of two options: *keep* or *remove*—the *choose* function selects one of the possible operations per symbol.

Line 11 then calls a subroutine that simply applies the chosen operations to the input KB. The following shows how the algorithm is applied in the running example.

Example 4.1. Consider the scenario from the running example, where we focus on how Dan reacts to the news items he sees in his feed (the ones from Example 2.2); recall that his local knowledge base is currently $\{\neg\beta, \neg\delta, \eta\}$.

According to the algorithm in Figure 10, the first step is to build the *collP* structure shown in Figure 12. The *conflict pattern* is identified for each predicate symbol; β's pattern is T_{13}, while for the rest we have T_4 (α), T_{13} (γ), T_{10} (δ), T_6 (ϵ), and T_{15} (ζ). Since the symbol β belongs to Dan's KB, *getSkylines-rel* is called and the possible operations are obtained (the skylines associated with T_{13}^{1} to T_{13}^{8} from Figure 13), and one of them has to be chosen and stored in *outcome*.

[4]For *restrained* operators, the only change required is the removal of the cases marked with "($\star$)".

Algorithm *reviseForRestrained*

Inputs:

NKB $= (V, E, l_{vert}, l_{edge}, K)$:	Network knowledge base	P :	Set of news items.
v:	Vertex in V.		
userType:	*restrained*/ *weakly_restrained*		

Output: NKB$'$, which includes the result of the revision of $K(v)$ by P.

```
1. outcome ← ∅ // Contains the chosen operations with respect to each pred. symb.
2. collP ← collapseNewsItems(P) // Map from pred. symb. to stateRecords, cf. Fig. 11
3. for each s ∈ keys(collP) do // Determine relevant SPOs-actions for each predicate symbol; cf. Fig. 7
4.   conflictPattern ← getInputPattern(s, collP)
5.   if s ∈ K(v) then
6.     outcome ← outcome ∪ {choose(getSkylines-rel(l, conflictPattern, userType))}
         // getSkylines-rel looks up conflictPattern in Figs. 13–14
7.   else
8.     outcome ← outcome ∪ {choose(getSkylines-unRel(l, conflictPattern, userType))}
         // getSkylines-unRel looks up conflictPattern in Fig. 15
9.   endif
10. endfor
11. K′(v) ← generateKB(K(v), outcome) // Build resulting KB w.r.t. the chosen operations.
12. K′ ← (K \ K(v)) ∪ {O}
13. return NKB′ = (V, E, l_vert, l_edge, K′).
```

Fig. 10. Algorithm for local NKB revision restrained and weakly restrained agents.

```
collP: Array(
    [pred-symbol_1] → stateRecord
    (
        [numPosAdd] → value_1
        [numPosRem] → value_2
        [numNegAdd] → value_3
        [numNegRem] → value_4
        [whoPosAdd] → PA
        [whoPosRem] → PR
        [whoNegAdd] → NA
        [whoNegRem] → NR
    )
    ...
    [pred-symbol_n] → stateRecord
    (
        [numPosAdd] → value_1
        [numPosRem] → value_2
        [numNegAdd] → value_3
        [numNegRem] → value_4
        [whoPosAdd] → PA
        [whoPosRem] → PR
        [whoNegAdd] → NA
        [whoNegRem] → NR
    )
)
```

Fig. 11. Structure for Algorithm in Figure 10: Collapsed news items.

Now consider γ, which is not in Dan's KB; its set of associated SPOs is obtained from T_{13}^{9}–T_{13}^{12}. The remaining predicate symbols are processed in a similar way; the following are the options for each one:

- α: $\{T_4^2\}$,
- β: $\{T_{13}^1, \ldots, T_{13}^8\}$,

```
collP:Array(
    [α] : stateRecord (
        [numPosAdd]  : 0
        [numPosRem]  : 0
        [numNegAdd]  : 0
        [numNegRem]  : 1
        [whoPosAdd]  : ∅
        [whoPosRem]  : ∅
        [whoNegAdd]  : ∅
        [whoNegRem]  : {Alice}
    )
    [β] : stateRecord (
        [numPosAdd]  : 1
        [numPosRem]  : 0
        [numNegAdd]  : 2
        [numNegRem]  : 1
        [whoPosAdd]  : {Beth}
        [whoPosRem]  : ∅
        [whoNegAdd]  : {Alice, Eric}
        [whoNegRem]  : {Fred}
    )
    [γ] : stateRecord (
        [numPosAdd]  : 2
        [numPosRem]  : 0
        [numNegAdd]  : 1
        [numNegRem]  : 1
        [whoPosAdd]  : {Beth, Fred}
        [whoPosRem]  : ∅
        [whoNegAdd]  : {Eric}
        [whoNegRem]  : {Alice}
    )
    [δ] : stateRecord (
        [numPosAdd]  : 0
        [numPosRem]  : 0
        [numNegAdd]  : 1
        [numNegRem]  : 1
        [whoPosAdd]  : ∅
        [whoPosRem]  : ∅
        [whoNegAdd]  : {Beth}
        [whoNegRem]  : {Cate}
    )
    [ϵ] : stateRecord (
        [numPosAdd]  : 3
        [numPosRem]  : 0
        [numNegAdd]  : 1
        [numNegRem]  : 0
        [whoPosAdd]  : {Alice, Beth, Cate}
        [whoPosRem]  : ∅
        [whoNegAdd]  : {Eric}
        [whoNegRem]  : ∅
    )
    [ζ] : stateRecord (
        [numPosAdd]  : 1
        [numPosRem]  : 1
        [numNegAdd]  : 1
        [numNegRem]  : 1
        [whoPosAdd]  : {Fred}
        [whoPosRem]  : {Beth}
        [whoNegAdd]  : {Eric}
        [whoNegRem]  : {Cate}
    )
)
```

Fig. 12. Data structure from Example 4.1.

- γ: $\{T_{13}^{9}, \ldots, T_{13}^{12}\}$,
- δ: $\{T_{10}^{1}, \ldots, T_{10}^{4}\}$,
- ϵ: $\{T_{6}^{5}, \ldots, T_{6}^{12}\}$, and
- ζ: $\{T_{15}^{33}, \ldots, T_{15}^{40}\}$.

For ease of presentation, here we used "T_Y^X" to denote the outcome associated with that SPO. The skylines of these SPOs encode actions that can be taken with respect to each predicate symbol; let us see in detail what these actions are for three of the symbols:

- For α, the possible skylines are N_4, N_3 (only valid for *weakly restrained* users) and N_1, which are interpreted as "*ignore* α and $\neg\alpha$", "*add* $\neg\alpha$", and "*add* α", respectively.
- The skylines for the symbol β (literal $\neg\beta$) are N_0, N_1, N_3, and N_4, which encode all possible operations: "*keep* $\neg\beta$", "*remove* $\neg\beta$", and "*add* β". Note that the operation"*remove* $\neg\beta$" is available only for the restrained case (i.e., it is not available for weakly-restrained revisions). This is because of SPO T_{13}^{6}, whose skyline is N_3, and is not allowed in the weakly restrained case.
- For γ, the nodes in the respective skylines encode the operations "*add* γ", "*add* $\neg\gamma$", and "*ignore* $\neg\gamma$". Note that the definition of "keep" and "ignore" depends on the status of the symbol in the KB.

According to line 6 and 8 in the algorithm, the function *choose* returns only one possible operation per symbol. Let us suppose that the *choose* function returns the following:

- Symbol α, given *userType*:
  - restrained:
    choose({"*ignore* α and $\neg\alpha$"}, "*add* α") $\rightarrow$ "*ignore* α and $\neg\alpha$"
  - weakly_restrained:
    choose({"*ignore* α and $\neg\alpha$"}, "*add* α", "*add* $\neg\alpha$") $\rightarrow$ "*add* $\neg\alpha$"
- Symbol β, given *userType*:
  - restrained:
    choose({"*keep* $\neg\beta$"}) $\rightarrow$ "*keep* $\neg\beta$"
  - weakly_restrained:
    choose({"*keep* $\neg\beta$", "*remove* $\neg\beta$", "*add* β"}) $\rightarrow$ "*add* β"
- Symbol γ: *choose*({"*ignore* γ and $\neg\gamma$", "*add* γ", "*add* $\neg\gamma$" }) $\rightarrow$ "*add* $\neg\gamma$"
- Symbol δ: *choose*({"*keep* $\neg\delta$", "*remove* $\neg\delta$", "*add* δ"}) $\rightarrow$ "*keep* $\neg\delta$"
- Symbol ϵ: *choose*({"*ignore* ϵ and $\neg\epsilon$", "*add* ϵ", "*add* $\neg\epsilon$"}) $\rightarrow$ "*ignore* ϵ and $\neg\epsilon$"
- Symbol ζ: *choose*({"*ignore* ζ and $\neg\zeta$", "*add* ζ", "*add* $\neg\zeta$"}) $\rightarrow$ "*add* ζ"

For the last four symbols we do not distinguish *userType*, since for both types the possible operations are the same in this example.

In summary, the chosen operations by *userType* are as follows:

- *restrained*:
  outcome = {"*ignore* α and $\neg\alpha$", "*keep* $\neg\beta$", "*add* $\neg\gamma$", "*keep* $\neg\delta$", "*ignore* ϵ and $\neg\epsilon$", "*add* ζ"}. Then, the revised KB is generated from $K(Dan)$ and the *outcome* set (line 11 in the algorithm), thus arriving at $K'(Dan) = \{\neg\beta, \neg\gamma, \neg\delta, \zeta, \eta\}$.
- *weakly_restrained*:
  outcome = {"*add* $\neg\alpha$", "*add* β", "*add* $\neg\gamma$", "*keep* $\neg\delta$", "*ignore* ϵ and $\neg\epsilon$", "*add* ζ"}. Then, $K'(Dan) = \{\neg\alpha, \beta, \neg\gamma, \neg\delta, \zeta, \eta\}$.

■

We now present our main representation theorem, which establishes a link between the theoretical characterization (set of postulates) and the construction of the operators (general algorithmic characterization). The value of this kind of result lies in that if future alternative constructions of the same operators are developed, then they can be shown to be equivalent by proving that they satisfy the same postulates.

Theorem 1. *Let* NKB $= (V, E, l_{vert}, l_{edge}, K)$ *be a network knowledge base,* $P = \{p_1, \ldots, p_n\}$ *be set of news items, and* $v \in V$ *and restrained/weakly restrained local NKB operator* $\circledast$. *Then,* NKB$'$ $= \circledast($NKB$, v, P)$ *if and only if* NKB$'$ *is a possible output of Algorithm* reviseForRestrained.

Proof. The proof consists of two parts:

Construction to postulates. We need to show that any revision that is produced by Algorithm *reviseForRestrained* satisfies both the basic postulates (Structural Preservation, Local Effect, Consistency, Uniformity, Inclusion and Congruence) as well as those for (weakly) restrained operators: (Weak) Vacuity 1 and 2.

One key observation to make is that the revised KB is taken from a pool of possibilities that is essentially determined by the SPOs in the table from Figures 13–15—this table was built by first considering all possible ways in which the graphs from Figure 7 can be oriented, and then removing from consideration those that either contain cycles or do not satisfy the *Inclusion* or *Vacuity* postulates. The rest of the postulates are easily shown to be satisfied:

Consistency: To guarantee *Consistency*, the algorithm first establishes conflicts via undirected edges between contradictory literals and between those literals with different associated decisions. Eventually, only one of those decisions can make it into the resulting KB.

Congruence and *Uniformity*: These postulates hold trivially in the setting considered in this article, since we only consider literals.

Structural Preservation and *Local Effect*: Since the algorithm does not modify any vertex or edge, and it only applies over a single local KB as input, these postulates are satisfied.

Therefore, any output produced by the algorithm satisfies the required postulates.

Postulates to construction. We first prove the following lemma:

Lemma 1. *Given the setup in Theorem 1, for each predicate symbol s appearing in an element from P_v, if $K'(v)$ is the result of an arbitrary revision then there is at least one summarized predicate symbol graph G such that there exists an orientation G' of G where the skyline nodes of G' characterize the operations that transform the status of s in $K(v)$ into the one in $K'(v)$.*

Proof (Of The Lemma). For any revision $K'(v)$ of $K(v)$ with respect to a predicate symbol s and set of news items P_v, there exists a set of operations that when applied to $K(v)$ we arrive at $K'(v)$—these operations can be one of *ignore*, *add*, or *remove* for s. Therefore, it is clear that the revision that yields $K'(v)$ from $K(v)$ for symbol s with respect to the corresponding news items in P_v can be characterized as an oriented summarized literal graph where the corresponding skyline yields the prescribed revision with respect to that literal, and the statement follows. □

We now need to show that all possible local revisions that satisfy all postulates for restrained/weakly restrained operators are generated by our algorithm, and therefore could be chosen as its output. The same argument used in the previous part of the proof can be used to show that this is the case: the algorithm chooses its output from the outcomes in the table in Figures 13–15. Since by Lemma 1 any revision can be mapped to at least one SPO arising from the orientation of a summarized predicate symbol graph (for the relevant predicate symbols), and considering that these tables were built by exhaustively considering all possible summarized literal graph patterns in Figure 7 and, for each one, all possible orientations, the property follows. □

Finally, we present the theoretical running time and space of our algorithm.

Theorem 2. *Let* NKB $= (V, E, l_{vert}, l_{edge}, K)$ *be a network knowledge base,* $P = \{p_1, \ldots, p_n\}$ *be set of news items, and* $v \in V$ *and restrained local NKB operator* $\circledast$. *Then, Algorithm* reviseForRestrained *obtains a revised NKB in time* $O(|P|)$*; the data structures used by the algorithm use space* $O(|P|)$.

Proof. The collapsed news items can be obtained in a single pass of set P (line 2). The `for` loop in lines 3–10 is executed at most $O(|P|)$ times, since there can be at most one distinct symbol per news item. In each iteration, the algorithm accesses the tables in Figures 13–15 ($O(1)$ time) to choose the outcome for the symbol in question. Finally, using adequate data structures, the operations in lines 11–13 can be carried out in time $O(1)$ with respect to the size of the algorithm's input structures. Regarding space, note that the space used by the algorithm is dominated by the *collP* data structure, which in the worst case has size $|P|$ and is then traversed only once. □

Note that the *choose* operation, which is carried out for each symbol in lines 6 and 8, could instead be done at the end; in this case, the algorithm would need to compute a set of possible outcomes (one for each possible operation), and then choose the resulting KB. Though this would afford a global choice to be made, the running time of the algorithm would jump from $O(|P|)$ to $O(3^{|P|})$, since each symbol potentially has three operations associated with it.

5 RELATED WORK

We now discuss several approaches that are related to the work presented in this article; though there are several research lines that are relevant for our work, to the best of our knowledge this is the first line of research to effectively take a belief revision approach to the problem of modeling knowledge dynamics in multiple social networks. As mentioned in Section 1, the underlying generalized network model can be seen as a *multi-layer* or *multiplex network*, and therefore our NKB model can be seen as an extension of this well-studied, pervasive model; we refer the interested reader to References [7, 37] for recent comprehensive surveys on this subject.

The two closest areas are belief dynamics in general, which we present in Section 5.1, and belief dynamics in social media contexts, which is the topic of Section 5.2. Finally, in Section 5.3 we discuss several other works that do not specifically fall within these two general areas but are nonetheless connected to our work.

5.1 Belief Dynamics

The problem of modeling how **knowledge bases (KBs)** change in response to different kinds of events is commonly known as *belief dynamics*; in particular, deciding how to integrate an epistemic input into a knowledge base is called *belief revision*. This problem has been studied mainly from two points of view: when KBs are comprised of formulas that are closed under logical consequence (called belief sets) [1, 27] and when they are not, in which case they are called belief bases [30, 32]. For a general survey on this topic, see Reference [50].

For our setting, one of the most closely related problems is that of *belief merging* [38, 39, 42] refers to the task of coherently combining *several* sources of information at once. During this process, new evidence can be partially or completely ignored if old information has more epistemic value; the operation integrates all the information into a consistent whole, whose result depends on informational value and not necessarily on novelty. A considerable amount of work has been developed in this area, in particular regarding theoretical studies. In Reference [39], an axiomatic characterization of merging operators is proposed and the operators that satisfy those axioms are called *pure merging operators*; later, Reference [38] considered the problem of merging several belief bases in the presence of integrity constraints, showing that the commutative revision operators from Reference [42] are a particular class of merging operators. Another line of work is related to *majority* and *arbitration* operators; the former is the class of operators in which a piece of information will persist in the final result of the merging if it came from many sources, while the latter focus on achieving consensus among the different sources of information. Later, this work was extended in Reference [10] by dealing with merging operators in an infinite logical framework (with countably many propositional variables)—their main result is a representation theorem where certain postulates had to be redefined to be appropriate for the infinite case. Also related to these works is that of Reference [17], where the use of a unanimity condition is approached, i.e., the acceptance of pieces of information based on the fact that it is shared by all the agents involved in the process.

Multiple change belief revision operators are also related to belief merging; in this case, the epistemic input is a set instead of a single sentence, and this approach is thus useful when combining different information sources. According to Reference [18], there are different types of multiple change operators: those in which the epistemic input is fully accepted (*prioritized change*) [1], those in which the epistemic input could be partially accepted (*non-prioritized change*) [19, 20, 30, 34], and those in which input sentences could be either accepted nor rejected (*symmetric change*) [21, 38].

NKB Local Revision vs. Belief Merging: A Closer Look. Belief revision is the process within belief dynamics that seeks to take into account a new piece of information with respect to an agent's KB, where the main assumption is that the new information is somehow more reliable or more important than the old information. Therefore, the task is to insert the new information into the set of old beliefs without generating an inconsistency. Belief merging, however, considers multiple KBs and the objective is that of *judgment aggregation*, that is to define an aggregation procedure that preserves individual rationality at the collective level. In practical terms, it seeks to build a new KB that somehow represents a *consensus* among the inputs.

In our setting, the focus is on *individual* agents and therefore a single KB, and the epistemic input consists of a set of news items that the user comes into contact with. So, it can be seen as a form of multiple revision that is not necessarily prioritized (i.e, the new information may not be accepted or may not be accepted completely) and where the input can be itself inconsistent. Intuitively, the goal is not to build a consensus from all news items, but rather to revise the agent's KB in light of the incoming information and without generating inconsistencies with previous beliefs. Furthermore, while merging seeks a global consensus and belief revision focuses on minimal change, we seek to define optimal changes in terms of particular behavioral characteristics of the users.

In a sense, the operators we seek to define cannot be considered to be merging operators but are neither pure nor standard (multiple) revision operators. In Reference [23] we carried out an extensive analysis of how existing merging and revision operators could be combined to fit this setting and show that the resulting behavior is not general enough to capture the complexities of social interactions and different types of social media users (in terms of behavior). We thus focused on the development of a different set of postulates that we consider more appropriate (though some of them are either generalizations or particularizations of revision postulates). The operators we study in this work therefore correspond to different subsets of the proposed postulates, each characterizing the behavior of different types of users. In future work, we plan to generalize this initial set of operators to capture broader ranges of possible behaviors.

Towards Iterated Revision. Finally, another subarea of belief dynamics that is closely related to our general approach is that of iterated revision [13, 47], in which operations are not considered in isolation but rather as part of a sequence. In our setting, it is natural to pursue the development of operators with this capability, since the overall NKB revision process can be seen as a cycle where local revisions lead to global ones, and the process is then repeated (as shown in Figure 3). Since our construction yields a new NKB in which there exist precedences over the remaining items of the original NKB together with the new ones, then the new NKB is suitable for a new revision. The study of the additional postulates and representation theorems, as well as an extension of the underlying language, will be part of future work.

5.2 Belief Revision in Social Contexts

In previous work [22], we proposed a general model called *Social Knowledge Bases* and identified a series of desirable properties for systems that work with the kind of data that is produced by agents in social media environments. The unique combination of challenges in this setting involves, among others: (i) multiple attributes with different domain data types, (ii) multiple relations between agents, (iii) uncertainty, (iv) reasoning with agents' preferences, (v) dealing with groups of agents as agents in their own right, (vi) belief revision, (vii) cascading processes, and (viii) computational tractability constraints. This is clearly a very complex and general problem—in this article, we tackled a subset of point (vi), defining concrete operators for local revision in the case in which the language in local KBs is restricted to ground literals. As mentioned above, in Reference [23] we took the first steps in this line of research by analyzing how traditional belief dynamics operators could be leveraged toward solving this generalized knowledge integration problem. In References [25, 26] we provided a preliminary presentation of some of the postulates described in this work, and reported on some initial experimental results showing how this framework can be used to detect user types in the Twitter network—this initial work informed the construction presented in this article.

Several other works have tackled the problem of belief dynamics in environments related to social media, where many different issues arise due to the interaction among agents. In Reference [29], trust among users is determined by profile similarity features; as a case study they considered the *Filmtrust* social network, in which users rank movies. Profile features are used to predict the trust of the users according to their similarities with other profiles—this kind of information could be part of the input in our setting to determine the strength in relations between users. In **Belief Revision Games (BRGs)** [52] an approach similar in spirit to ours is taken, where the belief dynamics in a group of agents communicating with each other are studied. In BRGs, at each step every agent revises its own belief state by considering the beliefs of its acquaintances according to a graph representation. There are several differences that separate this approach from ours, both from the point of view of the basic model adopted as well as the problems studied. First, quite different from our model, BRGs assume that agents have access to their connections' KBs, so agents cannot choose what to share, or whether or not to be truthful in what they share; the concept of news item in NKBs, which captures the concept of post in a social platform, allows for both aspects to be taken into account. Second, the underlying graph in BRGs is not labeled; as we argued in the introduction, multilayer networks capture the combination of multiple social platforms into one model, and allow a wide range of revision behaviors to take place (such as filtering out or giving more priority to beliefs held by agents with specific values for certain attributes). Third, BRGs assume that all beliefs are either in place when the game begins or arise as the result of an interaction, whereas in NKBs we allow exogenous events to modify agents' local KBs (which models, for instance, beliefs formed by watching TV or speaking with people in real life). From the point of view of the problems studied, in BRGs the concept of acceptability of beliefs is studied from the point of view of the result of the interactions throughout the game, and so the authors are more interested in problems such as studying conditions under which agents agree on some topic, or when convergence (no further changes occur) is achieved. Finally, BRGs adopt belief merging operators as the basis for all revisions, so their behavior is tied to the specific postulates proposed for that kind of operation. Notwithstanding all of these differences, we consider BRGs to be of interest for the future development and study of NKBs.

Cooperation in multi-agent systems also presents issues that can be tackled via coalitional games, which can be seen as an extension of BRGs [11, 16] where the players (agents) receive benefits when they engage in cooperation. In References [16, 65] a special class of coalitional games is presented, called *weighted voting games*, where each player has an associated weight and quota (threshold); when the total weight of the members participating in a coalition exceeds the quota, those players are considered to be *winners*. The *power* of a player in such a game could be determined, for instance, by well-known methods such as the Shapley-Shubik index [57] or the Banzhaf index [5]. These measures are important, since players can engage in manipulation, for instance toward reducing the power of another group [65]. In our approach, such direct manipulations cannot occur since the full graph structure is not known to each individual agent, and once news items are generated they cannot be taken back—studying other ways in which agents can engage in manipulation is, however, an interesting topic for future work. Similar to power and potential manipulation, *influence* has been studied in this kind of setting [43].

Belief negotiation models are introduced in Reference [8], which is a framework for merging information from *two* different sources—the pieces of information are weakened incrementally via a negotiation process until "common ground" is reached, i.e., until they become consistent with each other. Then, in Reference [9] that framework is extended by defining two families of *social contraction* functions, and is also generalized to n sources of information. Another related line of work is presented in Reference [59], where belief revision is also studied in the multi-agent systems setting; the authors consider the credibility or trust associated with each agent (referred to as *informants*), which is represented as a strict partial order among them. They define different kinds of change operators (expansion, contraction, and both prioritized and non-prioritized revision), and each operator is also able to modify the informant credibility relation as a response to new perceptions.

In References [53, 54], the distinction is made between symmetric and asymmetric relations in social media, which are also assumed to change over time. An epistemic logic of communities is developed in Reference [53], and different types of operators to infer distributed knowledge and relations are investigated—i.e., a query answering approach for knowledge in social media. A similar approach is taken in Reference [54], where a precise language for exploring "logic in the community" is developed and a two-dimensional modal logic is defined that allows to reason about the changing patterns of knowledge and social relationships in networks, on the basis of symmetric friendship relations.

Another important problem in belief dynamics in social settings is that of achieving *minimal change* with respect to the agent's previous beliefs. In Reference [15], a general framework for minimization-based belief change is proposed; their approach models general domains by considering a set of connected points with data associated with each one—this approach is therefore close in spirit to our work, but the model is different, since the set of points is structured in a simple graph with a single formula attached to each point. The revision operator determines a formula at each point by applying a minimization process; each new formula represents the integration between the original information associated with each point and the information coming from its related points. The notion of minimal change—similarly to Reference [28]—is given by the "closeness" among interpretations. Extensions are also defined to allow weights or reliability to be added to edges, and to establish a preference relation over interpretations.

Belief dynamics can also be seen as a "negotiation" process among agents to decide the result of the belief change operation for each agent. Conciliation operators are defined to characterize how agents' beliefs evolve according to an *iterated merge-then-revise* approach in Reference [28]. In that setting, an agent's new beliefs are obtained by confronting its previous beliefs. Two extreme ways to deal with this situation consists of giving more priority to previous beliefs (sceptical agents), or to give more importance to the beliefs of the group (credulous agents). In that work, the authors assume that the same revision operator is used by all the agents.

5.3 Other Approaches Related to Social Networks

Complex networks have been investigated in many different fields; for instance, in *decision making in group contexts* multiple experts are assumed to be expressing their opinions and a decision must be made to reach a common solution. The inconsistency problem caused by disparate opinions is addressed in Reference [44]; their aim is to generate customized advice for inconsistent experts to achieve consensus by considering informant trust levels. Then, in Reference [62] a visual consensus model is introduced to identify inconsistency and trust between unrelated users, which includes visual identification of the preference values with respect to a given threshold, advice generated by a recommendation simulation to increase current consensus, and the presentation of possible consequences if experts do not accept the recommended preference values. Other related problems addressed in complex networks in general are those of fake identity detection [12, 60] and sentiment analysis (a broad topic, for recent surveys see References [3, 49]).

Similarly to the works mentioned in the previous section, a critical factor in some group decision making environments is to respect the opinion of the majority of the participants; this approach is taken in Reference [4], where the authors define the Induced Ordered Weighted Averaging operator, which is a generalization of the OWA operator from Reference [63]. This operator uses an induced guiding principle to classify the outcome of several opinions into an aggregation mechanism. Also as mentioned above, another important aspect in these settings is that of trust and influence; Reference [61] proposes a mechanism to calculate trust values between users that do not know each other. In Reference [36], a model of trust that depends on the suitability with respect to a certain domain is proposed by defining state partitions. They define a family of trust-sensitive revision operators that are captured by the class of selective revision operators from Reference [20]. Before the revision, a trust evaluation process is carried out, and thus only trusted information is considered in that revision.

Finally, *social choice theory* [35] is quite relevant to our setting, since it deals with the identification, analysis, and evaluation of rules that can be used to make a collective decision. The intersection between this theory and computer science, called *computational social choice* [51], has been studied for several decades, and has applications in many real-world domains. The main difference between our approach and computational social choice is that local NKB revisions are "personal" operations, and therefore the agent that is carrying out revision is dictatorial—i.e., it can be seen as being open to opinions and suggestions, but they have the final say with respect to their own KB. Of course, different postulates could constrain the outcome of the operation, causing it to behave more like a social choice mechanism.

6 CONCLUSIONS AND FUTURE WORK

In this article, we have continued work on a general framework for modeling belief dynamics in social domains, called *Network Knowledge Bases*; though the model and preliminary results were presented in previous work, this is the first full formal treatment of the basic model, postulates, construction, and representation theorem. We have shown how restricting the language in local KBs to ground literals affords a construction based on enumerating orientations of base conflict graphs, and choosing the decisions independently for each predicate symbol. Current and future work is focused on generalizing the language to more complex formulas—we expect the approach to generalize beyond the current language, but complications to arise due to interactions between beliefs.

Another important topic of future work is to continue the empirical evaluations that we started in Reference [25], which are limited to identifying user types. With an algorithmic characterization now developed, we now wish to adapt it to real-world scenarios, and using the instantiated algorithms to solve problems such as behavior prediction, fake identity detection, and modeling complex processes such as cascades.

APPENDIX

A ADDITIONAL MATERIAL

Figures 13 and 14 show all the possible ways in which a summarized predicate symbol graph (which is undirected) can be oriented in the case in which a literal with the corresponding predicate symbol is already present in the KB; we use the following notation:

- $N_0 = (\alpha \in K(v))$; in the "skyline" column, stands for *keep* α.
- $N_1 = (\alpha, a, collP.\alpha.numPosAdd)$, stands for *keep* α.
- $N_2 = (\alpha, r, collP.\alpha.numPosRem)$, stands for *remove* α.
- $N_3 = (\neg\alpha, a, collP.\alpha.numNegAdd)$, and stands for *add* $\neg\alpha$

SPO	Conflict Pattern					Directed Edges in Graph	Skylines
	$K(v)$	*collP* (Input Pattern)					
	N_0	N_1	N_2	N_3	N_4		
T_1^1	✓	✓				$\emptyset$	N_0 , N_1
T_2^1	✓		✓			$\{e_{0\to 2}\}$	N_0
T_2^2	✓		✓			$\{e_{2\to 0}\}$	N_2
T_3^1	✓			✓		$\{e_{0\to 3}\}$	N_0
T_3^2	✓			✓		$\{e_{3\to 0}\}$	N_3
T_4^1	✓				✓	$\emptyset$	N_0 , N_4
T_5^1	✓	✓	✓			$\{e_{0\to 2}, e_{2\to 1}\}$	N_0
T_5^2	✓	✓	✓			$\{e_{0\to 2}, e_{1\to 2}\}$	N_0 , N_1
T_5^3	✓	✓	✓			$\{e_{2\to 0}, e_{2\to 1}\}$	N_2
T_5^4	✓	✓	✓			$\{e_{2\to 0}, e_{1\to 2}\}$	N_1
T_6^1	✓	✓		✓		$\{e_{0\to 3}, e_{3\to 1}\}$	N_0
T_6^2	✓	✓		✓		$\{e_{0\to 3}, e_{1\to 3}\}$	N_0 , N_1
T_6^3	✓	✓		✓		$\{e_{3\to 0}, e_{3\to 1}\}$	N_3
T_6^4	✓	✓		✓		$\{e_{3\to 0}, e_{1\to 3}\}$	N_0 , N_1
T_7^1	✓	✓			✓	$\emptyset$	N_0 , N_1 , N_4
T_8^1	✓		✓	✓		$\{e_{0\to 2}, e_{0\to 3}\}$	N_0
T_8^2	✓		✓	✓		$\{e_{0\to 2}, e_{3\to 0}\}$	N_3
T_8^3	✓		✓	✓		$\{e_{2\to 0}, e_{0\to 3}\}$	N_2
T_8^4	✓		✓	✓		$\{e_{2\to 0}, e_{3\to 0}\}$	N_2 , N_3
T_9^1	✓		✓		✓	$\{e_{0\to 2}\}$	N_0 , N_4
T_9^2	✓		✓		✓	$\{e_{2\to 0}\}$	N_2 , N_4
T_{10}^1	✓			✓	✓	$\{e_{0\to 3}, e_{4\to 3}\}$	N_0 , N_4
T_{10}^2	✓			✓	✓	$\{e_{0\to 3}, e_{3\to 4}\}$	N_0
T_{10}^3	✓			✓	✓	$\{e_{3\to 0}, e_{4\to 3}\}$	N_4
T_{10}^4	✓			✓	✓	$\{e_{3\to 0}, e_{3\to 4}\}$	N_3
T_{11}^1	✓	✓	✓	✓		$\{e_{0\to 2}, e_{0\to 3}, e_{2\to 1}, e_{3\to 1}\}$	N_0
T_{11}^2	✓	✓	✓	✓		$\{e_{0\to 2}, e_{0\to 3}, e_{2\to 1}, e_{1\to 3}\}$	N_0
T_{11}^3	✓	✓	✓	✓		$\{e_{0\to 2}, e_{0\to 3}, e_{1\to 2}, e_{3\to 1}\}$	N_0
T_{11}^4	✓	✓	✓	✓		$\{e_{0\to 2}, e_{0\to 3}, e_{1\to 2}, e_{1\to 3}\}$	N_0 , N_1
T_{11}^5	✓	✓	✓	✓		$\{e_{0\to 2}, e_{3\to 0}, e_{2\to 1}, e_{3\to 1}\}$	N_3
T_{11}^6	✓	✓	✓	✓		$\{e_{0\to 2}, e_{3\to 0}, e_{2\to 1}, e_{1\to 3}\}$	*cycle*
T_{11}^7	✓	✓	✓	✓		$\{e_{0\to 2}, e_{3\to 0}, e_{1\to 2}, e_{3\to 1}\}$	N_3
T_{11}^8	✓	✓	✓	✓		$\{e_{0\to 2}, e_{3\to 0}, e_{1\to 2}, e_{1\to 3}\}$	N_1
T_{11}^9	✓	✓	✓	✓		$\{e_{2\to 0}, e_{0\to 3}, e_{2\to 1}, e_{3\to 1}\}$	N_2
T_{11}^{10}	✓	✓	✓	✓		$\{e_{2\to 0}, e_{0\to 3}, e_{2\to 1}, e_{1\to 3}\}$	N_2
T_{11}^{11}	✓	✓	✓	✓		$\{e_{2\to 0}, e_{0\to 3}, e_{1\to 2}, e_{3\to 1}\}$	*cycle*
T_{11}^{12}	✓	✓	✓	✓		$\{e_{2\to 0}, e_{0\to 3}, e_{1\to 2}, e_{1\to 3}\}$	N_1
T_{11}^{13}	✓	✓	✓	✓		$\{e_{2\to 0}, e_{3\to 0}, e_{2\to 1}, e_{3\to 1}\}$	N_2 , N_3
T_{11}^{14}	✓	✓	✓	✓		$\{e_{2\to 0}, e_{3\to 0}, e_{2\to 1}, e_{1\to 3}\}$	N_2
T_{11}^{15}	✓	✓	✓	✓		$\{e_{2\to 0}, e_{3\to 0}, e_{1\to 2}, e_{3\to 1}\}$	N_3
T_{11}^{16}	✓	✓	✓	✓		$\{e_{2\to 0}, e_{3\to 0}, e_{1\to 2}, e_{1\to 3}\}$	N_1
T_{12}^1	✓	✓	✓		✓	$\{e_{0\to 2}, e_{2\to 1}\}$	N_0 , N_4
T_{12}^2	✓	✓	✓		✓	$\{e_{0\to 2}, e_{1\to 2}\}$	N_0 , N_1 , N_4
T_{12}^3	✓	✓	✓		✓	$\{e_{2\to 0}, e_{2\to 1}\}$	N_2 , N_4
T_{12}^4	✓	✓	✓		✓	$\{e_{2\to 0}, e_{1\to 2}\}$	N_1 , N_4

Fig. 13. (*Part 1/2*) Summarized predicate symbol graphs and corresponding SPOs for the case in which there is a literal with the corresponding symbol in the local KB. The skylines marked with "(⋆)" are not valid for restrained operators. The expression $e_{i\to j}$ represents an edge from N_i to N_j. Non-SPOs (i.e., orientations that yield cycles) are shown in gray.

- $N_4 = (\neg\alpha, r, collP.\alpha.numNegRem)$, stands for *ignore* $\neg\alpha$, since it represents the removal of a literal that is not in the KB.

The SPOs and skylines marked with "(⋆)" should not be considered in the case of *restrained* operators, since the operations they encode do not satisfy the operator's postulates.

SPO	Conflict Pattern					Directed Edges in Graph	Skylines
	$K(v)$	*collP* (Input Pattern)					
	N_0	N_1	N_2	N_3	N_4		
T^1_{13}	✓	✓		✓	✓	$\{e_{0\to3},e_{3\to1},e_{4\to3}\}$	$T^1_{15},...,T^{32}_{15}$ $\left(T^{11}_{15}, T^{15}_{15}, T^{17}_{15},...,T^{20}_{15}, T^{25}_{15},...,T^{28}_{15}, T^{31}_{15}\right)(\star)$
T^2_{13}	✓	✓		✓	✓	$\{e_{0\to3},e_{3\to1},e_{3\to4}\}$	
T^3_{13}	✓	✓		✓	✓	$\{e_{0\to3},e_{1\to3},e_{4\to3}\}$	
T^4_{13}	✓	✓		✓	✓	$\{e_{0\to3},e_{1\to3},e_{3\to4}\}$	
T^5_{13}	✓	✓		✓	✓	$\{e_{3\to0},e_{3\to1},e_{4\to3}\}$	
T^6_{13}	✓	✓		✓	✓	$\{e_{3\to0},e_{3\to1},e_{3\to4}\}$	
T^7_{13}	✓	✓		✓	✓	$\{e_{3\to0},e_{1\to3},e_{4\to3}\}$	
T^8_{13}	✓	✓		✓	✓	$\{e_{3\to0},e_{1\to3},e_{3\to4}\}$	
T^1_{14}	✓		✓	✓	✓	$\{e_{0\to2},e_{0\to3},e_{4\to3}\}$	N_0 , N_4
T^2_{14}	✓		✓	✓	✓	$\{e_{0\to2},e_{0\to3},e_{3\to4}\}$	N_0
T^3_{14}	✓		✓	✓	✓	$\{e_{0\to2},e_{3\to0},e_{4\to3}\}$	N_4
T^4_{14}	✓		✓	✓	✓	$\{e_{0\to2},e_{3\to0},e_{3\to4}\}$	N_3
T^5_{14}	✓		✓	✓	✓	$\{e_{2\to0},e_{0\to3},e_{4\to3}\}$	N_2 , N_4
T^6_{14}	✓		✓	✓	✓	$\{e_{2\to0},e_{0\to3},e_{3\to4}\}$	N_2
T^7_{14}	✓		✓	✓	✓	$\{e_{2\to0},e_{3\to0},e_{4\to3}\}$	N_2 , N_4
T^8_{14}	✓		✓	✓	✓	$\{e_{2\to0},e_{3\to0},e_{3\to4}\}$	N_2 , N_3
T^1_{15}	✓	✓	✓	✓	✓	$\{e_{0\to2},e_{0\to3},e_{2\to1},e_{4\to3},e_{3\to1}\}$	N_0 , N_4
T^2_{15}	✓	✓	✓	✓	✓	$\{e_{0\to2},e_{0\to3},e_{2\to1},e_{4\to3},e_{1\to3}\}$	N_0 , N_4
T^3_{15}	✓	✓	✓	✓	✓	$\{e_{0\to2},e_{0\to3},e_{2\to1},e_{3\to4},e_{3\to1}\}$	N_0
T^4_{15}	✓	✓	✓	✓	✓	$\{e_{0\to2},e_{0\to3},e_{2\to1},e_{3\to4},e_{1\to3}\}$	N_0
T^5_{15}	✓	✓	✓	✓	✓	$\{e_{0\to2},e_{0\to3},e_{1\to2},e_{4\to3},e_{3\to1}\}$	N_0 , N_4
T^6_{15}	✓	✓	✓	✓	✓	$\{e_{0\to2},e_{0\to3},e_{1\to2},e_{4\to3},e_{1\to3}\}$	N_0 , N_1 , N_4
T^7_{15}	✓	✓	✓	✓	✓	$\{e_{0\to2},e_{0\to3},e_{1\to2},e_{3\to4},e_{3\to1}\}$	N_0
T^8_{15}	✓	✓	✓	✓	✓	$\{e_{0\to2},e_{0\to3},e_{1\to2},e_{3\to4},e_{1\to3}\}$	N_0 , N_1
T^9_{15}	✓	✓	✓	✓	✓	$\{e_{0\to2},e_{3\to0},e_{2\to1},e_{4\to3},e_{3\to1}\}$	N_4
T^{10}_{15}	✓	✓	✓	✓	✓	$\{e_{0\to2},e_{3\to0},e_{2\to1},e_{4\to3},e_{1\to3}\}$	$cycle$
T^{11}_{15}	✓	✓	✓	✓	✓	$\{e_{0\to2},e_{3\to0},e_{2\to1},e_{3\to4},e_{3\to1}\}$	N_3
T^{12}_{15}	✓	✓	✓	✓	✓	$\{e_{0\to2},e_{3\to0},e_{2\to1},e_{3\to4},e_{1\to3}\}$	$cycle$
T^{13}_{15}	✓	✓	✓	✓	✓	$\{e_{0\to2},e_{3\to0},e_{1\to2},e_{4\to3},e_{3\to1}\}$	N_4
T^{14}_{15}	✓	✓	✓	✓	✓	$\{e_{0\to2},e_{3\to0},e_{1\to2},e_{4\to3},e_{1\to3}\}$	N_1 , N_4
T^{15}_{15}	✓	✓	✓	✓	✓	$\{e_{0\to2},e_{3\to0},e_{1\to2},e_{3\to4},e_{3\to1}\}$	N_3
T^{16}_{15}	✓	✓	✓	✓	✓	$\{e_{0\to2},e_{3\to0},e_{1\to2},e_{3\to4},e_{1\to3}\}$	N_1
T^{17}_{15}	✓	✓	✓	✓	✓	$\{e_{2\to0},e_{0\to3},e_{2\to1},e_{4\to3},e_{3\to1}\}$	N_2 , N_4
T^{18}_{15}	✓	✓	✓	✓	✓	$\{e_{2\to0},e_{0\to3},e_{2\to1},e_{4\to3},e_{1\to3}\}$	N_2 , N_4
T^{19}_{15}	✓	✓	✓	✓	✓	$\{e_{2\to0},e_{0\to3},e_{2\to1},e_{3\to4},e_{3\to1}\}$	N_2
T^{20}_{15}	✓	✓	✓	✓	✓	$\{e_{2\to0},e_{0\to3},e_{2\to1},e_{3\to4},e_{1\to3}\}$	N_2
T^{21}_{15}	✓	✓	✓	✓	✓	$\{e_{2\to0},e_{0\to3},e_{1\to2},e_{4\to3},e_{3\to1}\}$	$cycle$
T^{22}_{15}	✓	✓	✓	✓	✓	$\{e_{2\to0},e_{0\to3},e_{1\to2},e_{4\to3},e_{1\to3}\}$	N_1 , N_4
T^{23}_{15}	✓	✓	✓	✓	✓	$\{e_{2\to0},e_{0\to3},e_{1\to2},e_{3\to4},e_{3\to1}\}$	$cycle$
T^{24}_{15}	✓	✓	✓	✓	✓	$\{e_{2\to0},e_{0\to3},e_{1\to2},e_{3\to4},e_{1\to3}\}$	N_1
T^{25}_{15}	✓	✓	✓	✓	✓	$\{e_{2\to0},e_{3\to0},e_{2\to1},e_{4\to3},e_{3\to1}\}$	N_2 , N_4
T^{26}_{15}	✓	✓	✓	✓	✓	$\{e_{2\to0},e_{3\to0},e_{2\to1},e_{4\to3},e_{1\to3}\}$	N_2 , N_4
T^{27}_{15}	✓	✓	✓	✓	✓	$\{e_{2\to0},e_{3\to0},e_{2\to1},e_{3\to4},e_{3\to1}\}$	N_2 , N_3
T^{28}_{15}	✓	✓	✓	✓	✓	$\{e_{2\to0},e_{3\to0},e_{2\to1},e_{3\to4},e_{1\to3}\}$	N_2
T^{29}_{15}	✓	✓	✓	✓	✓	$\{e_{2\to0},e_{3\to0},e_{1\to2},e_{4\to3},e_{3\to1}\}$	N_4
T^{30}_{15}	✓	✓	✓	✓	✓	$\{e_{2\to0},e_{3\to0},e_{1\to2},e_{4\to3},e_{1\to3}\}$	N_1 , N_4
T^{31}_{15}	✓	✓	✓	✓	✓	$\{e_{2\to0},e_{3\to0},e_{1\to2},e_{3\to4},e_{3\to1}\}$	N_3
T^{32}_{15}	✓	✓	✓	✓	✓	$\{e_{2\to0},e_{3\to0},e_{1\to2},e_{3\to4},e_{1\to3}\}$	N_1

Fig. 14. *Part 2/2* – Continued from Figure 13.

For the case of symbols that are not in the KB, the corresponding information is presented in Figure 15; the notation is as follows:

- N_0 is a $v_{inertia} = (l \in lit(P_v)$ s.t. $l \notin K(v))$, stands for *ignore* α.
- $N_1 = (\alpha, a, collP.\alpha.numPosAdd)$, stands for *add* α.
- $N_2 = (\alpha, r, collP.\alpha.numPosRem)$, stands for *ignore* α.
- $N_3 = (\neg\alpha, a, collP.\alpha.numNegAdd)$, stands for *add* $\neg\alpha$.
- $N_4 = (\neg\alpha, r, collP.\alpha.numNegRem)$, stands for *ignore* $\neg\alpha$.

SPO	Conflict Pattern					Directed Edges in Graph	Skylines
	$v_{inertia}$	*collP* (Input Pattern)					
	N_0	N_1	N_2	N_3	N_4		
T_1^2	✓	✓				$\{e_{0\to1}\}$	N_0
T_1^3	✓	✓				$\{e_{1\to0}\}$	N_1
T_2^3			✓			$\emptyset$	T_{11}^{17}, T_{11}^{18}, T_{11}^{19} and $T_{11}^{20}(\star)$
T_3^3	✓			✓		$\{e_{0\to3}\}$	N_0
T_3^4	✓			✓		$\{e_{3\to0}\}$	N_3
T_4^2					✓	$\emptyset$	T_{13}^{9}, $T_{13}^{10}(\star)$, T_{13}^{11} and T_{13}^{12}
T_5^5		✓	✓			$\{e_{2\to1}\}$	T_{11}^{17} to T_{11}^{20}
T_5^6		✓	✓			$\{e_{1\to2}\}$	
T_6^5	✓	✓		✓		$\{e_{0\to1},e_{0\to3},e_{3\to1}\}$	N_0
T_6^6	✓	✓		✓		$\{e_{0\to1},e_{0\to3},e_{1\to3}\}$	N_0
T_6^7	✓	✓		✓		$\{e_{0\to1},e_{3\to0},e_{3\to1}\}$	N_3
T_6^8	✓	✓		✓		$\{e_{0\to1},e_{3\to0},e_{1\to3}\}$	*cycle*
T_6^9	✓	✓		✓		$\{e_{1\to0},e_{0\to3},e_{3\to1}\}$	*cycle*
T_6^{10}	✓	✓		✓		$\{e_{1\to0},e_{0\to3},e_{1\to3}\}$	N_1
T_6^{11}	✓	✓		✓		$\{e_{1\to0},e_{3\to0},e_{3\to1}\}$	N_3
T_6^{12}	✓	✓		✓		$\{e_{1\to0},e_{3\to0},e_{1\to3}\}$	N_1
T_7^2		✓			✓	$\emptyset$	T_{13}^{9}, $T_{13}^{10}(\star)$, T_{13}^{11} and T_{13}^{12}
T_8^5			✓	✓		$\emptyset$	T_{11}^{17}, T_{11}^{18}, T_{11}^{19} and $T_{11}^{20}(\star)$
T_9^3			✓		✓	$\emptyset$	N_2 , N_4
T_{10}^5				✓	✓	$\{e_{4\to3}\}$	T_{13}^{9} to T_{13}^{12}
T_{10}^6				✓	✓	$\{e_{3\to4}\}$	
T_{11}^{17}		✓	✓	✓		$\{e_{2\to1},e_{3\to1}\}$	N_2 , N_3
T_{11}^{18}		✓	✓	✓		$\{e_{2\to1},e_{1\to3}\}$	N_2
T_{11}^{19}		✓	✓	✓		$\{e_{1\to2},e_{3\to1}\}$	N_3
T_{11}^{20}		✓	✓	✓		$\{e_{1\to2},e_{1\to3}\}$	N_1
T_{12}^5		✓	✓		✓	$\{e_{2\to1}\}$	N_2 , N_4
T_{12}^6		✓	✓		✓	$\{e_{1\to2}\}$	N_1 , N_4
T_{13}^9		✓		✓	✓	$\{e_{3\to1},e_{4\to3}\}$	N_4
T_{13}^{10}		✓		✓	✓	$\{e_{3\to1},e_{3\to4}\}$	N_3
T_{13}^{11}		✓		✓	✓	$\{e_{1\to3},e_{4\to3}\}$	N_1 , N_4
T_{13}^{12}		✓		✓	✓	$\{e_{1\to3},e_{3\to4}\}$	N_1
T_{14}^9			✓	✓	✓	$\{e_{4\to3}\}$	N_2 , N_4
T_{14}^{10}			✓	✓	✓	$\{e_{3\to4}\}$	N_2 , N_3
T_{15}^{33}		✓	✓	✓	✓	$\{e_{2\to1},e_{4\to3},e_{3\to1}\}$	N_2 , N_4
T_{15}^{34}		✓	✓	✓	✓	$\{e_{2\to1},e_{4\to3},e_{1\to3}\}$	N_2 , N_4
T_{15}^{35}		✓	✓	✓	✓	$\{e_{2\to1},e_{3\to4},e_{3\to1}\}$	N_2 , N_3
T_{15}^{36}		✓	✓	✓	✓	$\{e_{2\to1},e_{3\to4},e_{1\to3}\}$	N_2
T_{15}^{37}		✓	✓	✓	✓	$\{e_{1\to2},e_{4\to3},e_{3\to1}\}$	N_4
T_{15}^{38}		✓	✓	✓	✓	$\{e_{1\to2},e_{4\to3},e_{1\to3}\}$	N_1 , N_4
T_{15}^{39}		✓	✓	✓	✓	$\{e_{1\to2},e_{3\to4},e_{3\to1}\}$	N_3
T_{15}^{40}		✓	✓	✓	✓	$\{e_{1\to2},e_{3\to4},e_{1\to3}\}$	N_1

Fig. 15. Summarized predicate symbol graphs and corresponding SPOs for the case in which there is no literal with the corresponding symbol in the local KB. The skylines marked with "($\star$)" are not valid for restrained operators. The expression $e_{i\to j}$ represents an edge from N_i to N_j. Non-SPOs (i.e., orientations that yield cycles) are shown in gray.

Note that some SPOs in the table redirect to others (*skyline* column) (T_2^3, T_4^2, T_5^5, T_5^6, T_7^2, T_8^5, T_{10}^5, T_{10}^6, and T_{13}^1–T_{13}^8). This allows for the possibility of considering the addition of a literal l or $\neg l$ to the KB in cases in which the symbol associated symbol only appears in the news item set with *remove* as decision; otherwise, given the interpretation assigned to nodes N_0–N_4 in the skylines, this operation could not be considered unless a news item contains the literal (or its negation) with an *add* as decision. These operations need to be taken into account because some postulates consider the possibility of adding a symbol that only has removals as decision in the feeds (e.g., *Weak Vacuity 1*). Therefore, some SPOs are redirected to those that include the original nodes plus the nodes N_1 and N_3 (which are interpreted as add l and add $\neg l$, respectively). However, to determine the valid SPOs we must consider the original ones created arising from the news item set and KB.

For instance, SPO T_7^2 in Figure 15—whose *skylines* column redirects to T_{13}^{10} ($\star$), among others—the skyline for SPO T_{13}^{10} contains the node N_3, which is interpreted as "*add* $\neg l$" to the KB; since this operation does not satisfy *Vacuity 1*, it is not valid for *restrained* operators (cf. Definition 3.3). In other words, SPO T_7^2 for a restrained operator is redirected to those that contain the nodes N_1 and N_4 representing the news item set that has at least a news item adding l and one removing $\neg l$, and there is no other operation in the set referred to the symbol. However, the skyline T_{13}^{10} is only valid for *weakly restrained* operators, since it represents the possibility of adding $\neg l$ to the KB; this operation does not satisfy the *Vacuity 1* postulate's second condition. Informally, this postulate establishes that given a symbol that is not in the KB, if every news item associated with the literal $\neg l$ implies a *remove* decision (which is the case for SPO T_7^2), then the literal cannot be included in the revised KB.

REFERENCES

[1] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. 1985. On the logic of theory change: Partial meet contraction and revision functions. *J. Symbol. Logic* 50, 2 (1985), 510–530.

[2] Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *J. Econ. Perspect.* 31, 2 (2017), 211.

[3] Orestes Appel, Francisco Chiclana, and Jenny Carter. 2015. Main concepts, state of the art and future research questions in sentiment analysis. *Acta Polytechn. Hung.* 12, 3 (2015), 87–108.

[4] Orestes Appel, Francisco Chiclana, Jenny Carter, and Hamido Fujita. 2017. A consensus approach to the sentiment analysis problem driven by support-based IOWA majority. *Int. J. Intell. Syst.* 32, 9 (2017), 947–65.

[5] John F. Banzhaf III. 1964. Weighted voting doesn't work: A mathematical analysis. *Rutgers L. Rev.* 19 (1964), 317.

[6] Alain Barrat, Marc Barthelemy, Romualdo Pastor-Satorras, and Alessandro Vespignani. 2004. The architecture of complex weighted networks. *Proc. Natl. Acad. Sci. U.S.A.* 101, 11 (2004), 3747–3752.

[7] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I. Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin. 2014. The structure and dynamics of multilayer networks. *Phys. Rep.* 544, 1 (2014), 1–122.

[8] Richard Booth. 2001. A negotiation-style framework for non-prioritised revision. In *TARK*. Morgan Kaufmann Publishers Inc., 137–150.

[9] Richard Booth. 2006. Social contraction and belief negotiation. *Inf. Fus.* 7, 1 (2006), 19–34.

[10] José Luis Chacón and Ramón Pino Pérez. 2004. Logic-based merging: The infinite case. In *NMR*. 100–108.

[11] Vincent Conitzer and Tuomas Sandholm. 2006. Complexity of constructing solutions in the core based on synergies among coalitions. *Artif. Intell.* 170, 6–7 (2006), 607–619.

[12] Matteo Cristani, Elisa Burato, Katia Santacá, and Claudio Tomazzoli. 2015. The spider-man behavior protocol: Exploring both public and dark social networks for fake identity detection in terrorism informatics. In *KDWeb*. 77–88.

[13] Adnan Darwiche and Judea Pearl. 1997. On the logic of iterated belief revision. *Artif. Intell.* 89, 1–2 (1997), 1–29.

[14] Marijke De Veirman, Veroline Cauberghe, and Liselot Hudders. 2017. Marketing through instagram influencers: the impact of number of followers and product divergence on brand attitude. *Int. J. Advert.* 36, 5 (2017), 798–828.

[15] James P. Delgrande, Jérôme Lang, and Torsten Schaub. 2007. Belief change based on global minimisation. In *IJCAI*, Vol. 7. 2468–2473.

[16] Edith Elkind, Leslie Ann Goldberg, Paul W. Goldberg, and Michael Wooldridge. 2009. On the computational complexity of weighted voting games. *Ann. Math. Artif. Intell.* 56, 2 (2009), 109–131.

[17] Patricia Everaere, Sébastien Konieczny, and Pierre Marquis. 2010. Disjunctive merging: Quota and gmin merging operators. *Artif. Intell.* 174, 12 (2010), 824–849.

[18] Marcelo A. Falappa, Gabriele Kern-Isberner, Maurício D. L. Reis, and Guillermo R. Simari. 2012. Prioritized and non-prioritized multiple change on belief bases. *J. Philos. Logic* 41, 1 (2012), 77–113.

[19] Marcelo A. Falappa, Gabriele Kern-Isberner, and Guillermo R. Simari. 2002. Belief revision, explanations and defeasible reasoning. *Artif. Intell. J.* 141 (2002), 1–28.

[20] Eduardo L Fermé and Sven Ove Hansson. 1999. Selective revision. *Stud. Logic.* 63, 3 (1999), 331–342.

[21] André Fuhrmann. 1997. *An Essay on Contraction.* Studies in Logic, Language and Information. CSLI Publications, Stanford, CA.

[22] Fabio R. Gallo, Natalia Abad Santos, Gerardo I. Simari, and Marcelo A. Falappa. 2015. A desiderata for modeling and reasoning with social knowledge. In *CACIC.*

[23] Fabio R. Gallo, Natalia Abad Santos, Gerardo I. Simari, Maria Vanina Martinez, and Marcelo A. Falappa. 2016. Belief dynamics in complex social networks. In *ASAI'16.*

[24] Fabio R. Gallo, Gerardo I. Simari, Maria Vanina Martinez, and Marcelo A. Falappa. 2020. Predicting user reactions to Twitter feed content based on personality type and social cues. *Fut. Gener. Comput. Syst.* 110 (2020), 918–930.

[25] Fabio R. Gallo, Gerardo I. Simari, Maria Vanina Martinez, Marcelo A. Falappa, and Natalia Abad Santos. 2017. Reasoning about sentiment and knowledge diffusion in social networks. *IEEE Internet Comput.* 21, 6 (2017), 8–17.

[26] Fabio R. Gallo, Gerardo I. Simari, Maria Vanina Martinez, Natalia Abad Santos, and Marcelo A. Falappa. 2017. A first approach to belief dynamics in complex social networks. In *AMW (CEUR Workshop Proceedings)*, Vol. 1912.

[27] Peter Gärdenfors. 1988. *Knowledge in Flux: Modeling the Dynamics of Epistemic States.* The MIT Press.

[28] Olivier Gauwin, Sébastien Konieczny, and Pierre Marquis. 2007. Conciliation through iterated belief merging. *J. Logic Comput.* 17, 5 (2007), 909–937.

[29] Jennifer Golbeck. 2009. Trust and nuanced profile similarity in online social networks. *ACM Trans. Web* 3, 4 (2009), 12.

[30] Sven Hansson. 1997. Semi-revision. *J. Appl. Non-Classic. Logics* 7, 1–2 (1997), 151–175.

[31] Sven Ove Hansson. 1993. Reversing the Levi identity. *J. Philos. Logic* 22, 6 (1993), 637–669.

[32] Sven Ove Hansson. 1994. Kernel contraction. *J. Symbol. Logic* 59, 3 (1994), 845–859.

[33] Sven Ove Hansson. 1999. *A Textbook of Belief Dymanics: Theory Change and Database Updating.* Kluwer Academic Publishers.

[34] Sven Ove Hansson, Eduardo Leopoldo Fermé, John Cantwell, and Marcelo A. Falappa. 2001. Credibility limited revision. *J. Symbol. Logic* 66, 4 (2001), 1581–1596.

[35] Jac C. Heckelman and Nicholas R. Miller. 2015. *Handbook of Social Choice and Voting.* Edward Elgar Publishing.

[36] Aaron Hunter and Richard Booth. 2015. Trust-sensitive belief revision. In *IJCAI.* 3062–3068.

[37] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. 2014. Multilayer networks. *J. Complex Netw.* 2, 3 (2014), 203–271.

[38] Sébastien Konieczny and Ramón Pino Pérez. 2002. Merging information under constraints: A logical framework. *J. Logic Comput.* 12, 5 (2002), 773–808.

[39] Sébastien Konieczny and Ramón Pino-Pérez. 1998. On the logic of merging. In *KR'98.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 488–498.

[40] Issie Lapowswky. 2016. Here's How Facebook Actually Won Trump the Presidency. Retrieved February 14, 2018 from https://www.wired.com/2016/11/facebook-won-trump-election-not-just-fake-news/

[41] K. Lewis, J. Kaufman, M. Gonzalez, A. Wimmer, and N. A. Christakis. 2008. Tastes, ties, and time: A new (cultural, multiplex, and longitudinal) social network dataset using facebook.com. *Soc. Netw.* 30 (2008), 330–342.

[42] Paolo Liberatore and Marco Schaerf. 1998. Arbitration (or how to merge knowledge bases). *IEEE Trans. Knowl. Data Eng.* 10, 1 (1998), 76–90.

[43] Fenrong Liu, Jeremy Seligman, and Patrick Girard. 2014. Logical dynamics of belief change in the community. *Synthese* 191, 11 (2014), 2403–2431.

[44] Yujia Liu, Changyong Liang, Francisco Chiclana, and Jian Wu. 2017. A trust induced recommendation mechanism for reaching consensus in group decision making. *Knowl.-Bas. Syst.* 119 (2017), 221–231.

[45] Akitaka Matsuo and Kenneth Benoit. 2017. More positive, assertive and forward-looking: How leave won Twitter. *LSE Brexit* (2017).

[46] Giulia Menichetti, Daniel Remondini, Pietro Panzarasa, Raúl J Mondragón, and Ginestra Bianconi. 2014. Weighted multiplex networks. *PloS One* 9, 6 (2014), e97857.

[47] Abhaya C. Nayak. 1994. Iterated belief change based on epistemic entrenchment. *Erkenntnis* 41, 3 (1994), 353–390.

[48] Brian L. Ott. 2017. The age of Twitter: Donald J. Trump and the politics of debasement. *Crit. Stud. Media Commun.* 34, 1 (2017), 59–68.

[49] Georgios Paltoglou and Mike Thelwall. 2017. Sensing social media: A range of approaches for sentiment analysis. In *Cyberemotions.* Springer, 97–117.

[50] Pavlos Peppas. 2008. Belief revision. *Found. Artif. Intell.* 3 (2008), 317–359.

[51] Francesca Rossi, Kristen Brent Venable, and Toby Walsh. 2011. A short introduction to preferences: between artificial intelligence and social choice. *Synth. Lect. Artif. Intell. Mach. Learn.* 5, 4 (2011), 1–102.

[52] Nicolas Schwind, Katsumi Inoue, Gauvain Bourgne, Sébastien Konieczny, and Pierre Marquis. 2015. Belief revision games.. In *AAAI*, Vol. 15. 1590–1596.

[53] Jeremy Seligman, Fenrong Liu, and Patrick Girard. 2011. Logic in the community. *Ind. Conf. Logic Appl.* 6521 (2011), 178–188.
[54] Jeremy Seligman, Fenrong Liu, and Patrick Girard. 2013. Facebook and the epistemic logic of friendship. In *TARK*.
[55] Paulo Shakarian, Matthias Broecheler, V. S. Subrahmanian, and Cristian Molinaro. 2013. Using generalized annotated programs to solve social network diffusion optimization problems. *ACM Trans. Comput. Logic* 14, 2 (2013), 10:1–10:40.
[56] Chengcheng Shao, Giovanni Luca Ciampaglia, Alessandro Flammini, and Filippo Menczer. 2016. Hoaxy: A platform for tracking online misinformation. In *WWW Companion*. 745–750.
[57] Lloyd S. Shapley and Martin Shubik. 1954. A method for evaluating the distribution of power in a committee system. *Am. Pol. Sci. Rev.* 48, 3 (1954), 787–792.
[58] Sławek Staworko, Jan Chomicki, and Jerzy Marcinkowski. 2012. Prioritized repairing and consistent query answering in relational databases. *Ann. Math. Artif. Intell.* 64, 2 (01 Mar. 2012), 209–246.
[59] Luciano H. Tamargo, Alejandro J. García, Marcelo A. Falappa, and Guillermo R. Simari. 2014. On the revision of informant credibility orders. *Artif. Intell.* 212 (2014), 36–58.
[60] Sai Krishna Vallapu. 2016. *Towards Network False Identity Detection in Online Social Networks*. Ph.D. Dissertation. Southern Illinois University at Edwardsville.
[61] Patricia Victor, Chris Cornelis, Martine De Cock, and Enrique Herrera-Viedma. 2011. Practical aggregation operators for gradual trust and distrust. *Fuzzy Sets Syst.* 184, 1 (2011), 126–147.
[62] Jian Wu, Francisco Chiclana, Hamido Fujita, and Enrique Herrera-Viedma. 2017. A visual interaction consensus model for social network group decision making with trust propagation. *Knowl.-Bas. Syst.* 122 (2017), 39–50.
[63] Ronald R Yager. 1988. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Trans. Syst. Man Cybernet.* 18, 1 (1988), 183–190.
[64] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing. *iIEEE Comput. Intell. Mag.* 13, 3 (2018), 55–75.
[65] Michael Zuckerman, Piotr Faliszewski, Yoram Bachrach, and Edith Elkind. 2008. Manipulating the quota in weighted voting games. In *AAAI*, Vol. 8. 215–220.

Received October 2019; revised February 2021; accepted July 2021