

Building a Network Knowledge Base Based on a Belief Revision Operator

Reynold Osuna-González and Guillermo De Ita-Luna(✉)

Benemérita Universidad Autónoma de Puebla, Puebla, México
deitaluna63@gmail.com

Abstract. We present a proposal for negotiation in a network knowledge base (*NKB*) formed by a set of k intelligent agents, in order to decide the acceptance (or not) of a new information ϕ that is expressed in propositional logic.

Each one of the agents Ai, $i = 1,\ldots,k$ has associated a knowledge base (*KB*) Ki expressed as propositional formulas in conjunctive form. In order to decide the acceptance of ϕ, the belief revision operator *Ind* is applied on each *KB* Ki and ϕ, then $Si = Ind(\phi, Ki)$, and each one of the literals that form Si is weighed in order to estimate the cost that is required by Ai to accept ϕ.

Si expresses the required information by the agent Ai such that $(Ki \cup Si) \models \phi$. The *NKB* uses a constant value γ that is the threshold of acceptance of *NKB*. Thus, a piece of new information ϕ will be accepted by all the agents in NKB if the average of the weights of the requested formulas Si is greater than γ. Otherwise, ϕ is rejected by all the agents in the *NKB*.

Only in the case that ϕ will be accepted collectively, each one of the Ki in the network *NKB* is updated, as $Ki = (Ki \bigwedge Si)$. Otherwise, there are no changes to any *KB* Ki in the *NKB*.

Keywords: Propositional · Inference Belief Revision · Network Knowledge Base

1 Introduction

The automation of deductive reasoning is a basic and challenging logic problem [1]. Deductive propositional reasoning is usually abstracted as follows: Given a propositional formula K (capturing the knowledge about a domain), and a propositional formula ϕ (a query capturing the situation at hand), then the goal of reasoning is to determine whether K implies ϕ, which is presented as $K \vDash \phi$. This is known as the propositional inference problem (or the entailment problem).

Since automatic reasoning is one of the purer forms of human intellectual thought, the automation of such reasoning by means of computers is a basic and challenging scientific problem [1]. Thus, one of the fundamental problems in automatic reasoning is the propositional inference problem. This last problem is a relevant task in many other issues such as: estimating the degree of belief, belief revision, abductive explanation, logical diagnosis, and many other procedures in Artificial Intelligence (AI) applications.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023 Workshops, LNCS 14104, pp. 3–14, 2023.
https://doi.org/10.1007/978-3-031-37105-9_1

Inference operations are present not only in logic but in many areas of mathematics. The concept of logical inference has proven to be more fruitful for the development of a general logic theory than the concept of theorem and of logical validity. Abstract inference operations are known as closure operators in universal algebra and lattice theory [2].

The current techniques that are most used to automatically check $K \vDash \phi$, when K and ϕ are conjunctive normal forms (CF), are the resolution method [3] and the systems type Gentzen [4]. However, these processes have not only an inherent exponential complexity, but also, they lack a recovery proposal when the inference does not hold. On the contrary, our proposal for checking $K \vDash \phi$ allows to build a new CF S for repairing the inference, such that $(S \cup K) \vDash \phi$. In our proposal, $(S \cup K)$ continues as a CF; therefore, our proposal is closed on conjunctive forms. If K represents a knowledge base (KB) associated with an intelligent agent A, then S expresses the required information by the agent A such that $(K \cup S) \models \phi$.

Belief revision problem consists in incorporate new beliefs to a knowledge base already established, changing as little as possible the original beliefs and maintaining consistency of the KB. We propose a method for belief revision, first considering a unique intelligent agent, and afterward, a network of intelligent agents that want to consider if new information is or not accepted in a collective form.

Although different types of logic have been used in decision-making processes by intelligent agents, such as modal logic, fuzzy logic, temporal logic, etc. In our case, we have circumscribed our work within propositional logic because our proposal for group decision-making of agents is based on using a belief revision operator, which we have already developed and analyzed their behavior and consistency in propositional logic [5]. Furthermore, we have extended our original proposal of a belief revision operator to be applied in the revision of consistency of a knowledge base [6].

People are becoming increasingly more connected to each other as social networks continue to grow both in number and variety [7]. Taking them as a collection, a network agents 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 agent. In this context, a network agents formed by intelligent agents 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 an introduction to the *network knowledge base* (*NKB*) model, and then focus on the problem of how a single agent's knowledge base changes when exposed to a stream of new items coming from the reality. Afterwards, we show how to make a collective decision of acceptance, if new information is considered for the network knowledge base.

2 Preliminaries

Let $X = \{ x_1, \ldots, x_n \}$ be a set of n Boolean variables. A literal, denoted as *lit,* is a variable x or a denied variable $\neg x$. As usual, each $x \in X$, $x^0 = \neg x$ and $x^1 = x$.

A *clause* is a disjunction of different literals. For $k \in N$, a *k-clause* is a clause with exactly k literals, and ($\leq k$)-*clause* is a clause with at most k literals. Sometimes, we consider a clause as a set of literals.

A *phrase* is a conjunction of literals. A *k-phrase* is a phrase with exactly k literals. A variable $x \in X$ appears in a clause C (or phrase) if x or $\neg x$ is an element of C.

A conjunctive normal form (*CF*) is a conjunction of clauses, and *k-CF* is a *CF* containing only *k-clauses*. A disjunctive normal form (*DF*) is a disjunction of sentences, and *k-DF* is a *DF* containing only *k-phrases*. A CF F with n variables is a *n-ary* Boolean function $F:\{0,1\}^n \rightarrow \{0,1\}$. Rather, any Boolean function F has infinitely many equivalent representations, among these, some in *CF* and *DF*.

We denote with Y any of the basic logic elements that we are using, such as a literal, a clause, a phrase, a *DF* or a C. $v(Y)$ denotes the set of variables involved in the object Y. For example, $Y(\neg x_1 \bigvee x_2) = \{x_1, x_2\}$. While $Lit(Y)$ denotes the set of literals involved in object Y. For example, if $X = v(Y)$ then $Lit(Y) = X \cup \neg X = \{x_1, \neg x_1, \ldots, x_n, \neg x_n\}$. We also use $\neg Y$ as the negation operator on the object Y. We denote the cardinality of a set A by $|A|$.

An *assignment s* for a formula F is a Boolean mapping $s: v(F) \rightarrow \{1, 0\}$. An assignment s can also be considered as a non-complementary set of literals: $l \in s$ if and only if s assigns true to l and $\neg l$ false. s is a partial *assignment* for the formula F when s has determined a logical value only to variables of a proper subset of F, namely $s: Y \rightarrow \{1, 0\}$ and $Y \subset v(F)$. A *CF* F is satisfied by an assignment s if each clause F is satisfied by s; A model of F is an assignment on $v(F)$ satisfying F. Considering s as a set of literals, notice that s has not any complementary pair of literals, this is, if $l \in s$, then $\neg l \notin s$.

A phrase f is satisfiable by an assignment s, if $f \subseteq s$. Otherwise, s falsifies f. A *DF* F is satisfiable by s if a sentence in F is satisfiable by s. F is contradicted by s if all sentences in F are contradicted by s.

Given a formula F, $S(F)$ is the set of all possible mappings defined on $v(F)$. If $n = |v(F)|$ then $|S(F)| = 2^n$. $s \vDash F$ denotes that the assignment s is a model of F (s satisfies F). $s \nvDash F$ denotes that s is an assignment falsifying F. We denote by $Sat(F)$ to the set of assignment in $S(F)$ which they are models for F. $Fals(F)$ denotes the set of assignments in $S(F)$ that falsifies F. Similarly, $\#Sat(F)$ and $\#Fals(F)$ denotes the number of models and falsifying assignments of F, respectively.

2.1 Construction of Independent Sets of Clauses

As K and ϕ are *CF*'s, the falsifying assignments $Fals(K)$ y $Fals(\phi)$ can be calculated efficiently [8]. Review $K \vDash \phi$ is equivalent to checking whether $Sat(K) \subseteq Sat(\phi)$, or that: $Fals(\phi) \subseteq Fals(K)$. The result of applying the operator of belief revision between a KB K and the new evidence ϕ is denoted as $K' = K \circ \phi$.

When $K \vDash \phi$, then we define $K = K \circ \phi$, because the new evidence is inferred from K, end then K does not need any information to accept ϕ. If $K \nvDash \phi$ then $Fals(\phi) \not\subset Fals(K)$, which implies that there is a set of assignments S such that $S \subseteq Fals(\phi)$ and $S \not\subseteq Fals(K)$.

When K $\nvDash \phi$, we propose a method to build $S = (Fals(\phi)$ - $Fals(K)) \neq \emptyset$ which allows to recognize a new *CF Fs*, such that $S = Fals(Fs)$ and $K' = (K \bigwedge Fs)$, and it

holds $K' \vDash \phi$. $K' = (K \bigwedge Fs)$ has less information than K (because K' has more clauses than K). In fact, if $S \neq 0$ then $Fals(K) \subset Fals(K')$, and therefore, $Sat(K') \subset Sat(K)$.

Let K be a CF, i.e., $K = \bigwedge_{i=1}^{m} C_i$, where each C_i, $i = 1, \ldots, m$ is a disjunction of literals. The set of assignments forming $Fals(C_i)$ can be represented through a string A_i from the set $\{0, 1, *\}$. If $C_i = \{x_{i1} \bigvee \ldots \bigvee x_{ik}\} \in K$, then the value for each position from i_1 - th to i_k – th of the string A_i, has to falsify the literals of C_i. If $x_{ij} \in C_i$ then the ij – th element of A_i is set to 0. If $\neg x_{ij} \in C_i$ then the ij – th element of A_i is set to 1. The variables in $v(K)$ which do not appear in C_i are represented by the symbol *, meaning that they could take any logical value $\{0, 1\}$. In this way, the string A_i of length $n = |v(K)|$ represents the set of assignments falsifying the clause C_i.

We will denote the formation of the string representing $Fals(C_i)$ as $A_i = \text{string}(Fals(C_i))$. It is easy to build $Fals(K)$ since each clause C_i determines a subset of falsifying assignments of K.

In this article, we will express $Fals(C)$ as the falsifying string or as the set of literals in an indistinct way, since both expressions denote the set $Fals(C)$. The following property expresses how to form the falsifying set of assignments of a CF.

Property 1. *Given a CF* $K = \bigwedge_{i=1}^{m} Ci$, *it holds* $Fals(K) = \cup_{i=1}^{m} \{\sigma \in S(K) \mid Fals(C_i) \subseteq \sigma\}$.

Definition 1. *Given two Clauses* C_i *and* C_j, *if they have at least one complementary literal, they will be called independent clauses. Otherwise, it is said that both are dependent clauses.*

Two independent clauses C_i and C_j have complementary pair of literals, therefore their falsifying assignments must also have complementary literals, that is, $Fals(C_i) \bigcap Fals(C_j) = \emptyset$.

Definition 2. *Let* $K = \{C_1, C_2, \ldots, C_m\}$ *be a CF. K is called independent if for any pair of clauses* $C_i, C_j \in K$, $i \neq j$, *the property of independence is met.*

Definition 3. *Given two falsifying strings A and B each of length n, if there is an* $i = 0,\ldots, n$, *such that* $A[i] = x$ *and* $B[i] = 1 - x$, $x \in \{0, 1\}$, *it is said that they have the* independence *property. Otherwise, we say that both strings are dependent.*

Notice that falsifying strings for independent clauses have complementary values (0 and 1) in at least one of their fixed values. Let $C_i \in K$ and $x \in v(K) \backslash v(C_i)$ be any variable, we have that

$$C_i \equiv (C_i \vee \neg x) \bigwedge (C_i \vee x) \tag{1}$$

Furthermore, this reduction preserves the number of falsifying assignments of C_i. Since $\#Fals(C_i) = 2^{n-|Ci|} = 2^{n-(|Ci|+1)} + 2^{n-(|Ci|+1)} = \#\text{Fals}((C_i \vee \neg x) \bigwedge (C_i \vee x))$, because $(C_i \vee x)$ and $(C_i \vee x)$ are two independent clauses.

Definition 4. *Given a pair of dependent clauses C_1 and C_2, if $Lit(C_1) \subseteq Lit(C_2)$ we say that C_2 is subsumed by C_1.*

If C_1 subsumes C_2 then $Fals(C_2) \subseteq Fals(C_1)$. On the other hand, if C_2 is not subsumed by C_1 and they are dependents, there is a set of indices $I = \{1, ..., p\} \subseteq \{1, ..., n\}$ such that for each $i \in I$, $x_i \in C_1$ but $x_i \notin C_2$. There exists a reduction to transform C_2 to be independent with C_1, we call this transformation the independent reduction between two clauses.

The independent reduction works as follows: let C_1 and C_2 be two dependent clauses. Let $\{x_1, x_2, ..., x_p\} = Lit(C_1) - Lit(C_2)$. By (1) we can write: $C_1 \bigwedge C_2 = C_1 \bigwedge (C_2 \vee \neg x_1) \bigwedge (C_2 \vee x_1)$. Now C_1 and $(C_2 \vee \neg x_1)$ are independent. Applying (1) to (C2 $\vee$ x_1):

$$C_1 \bigwedge C_2 \equiv C_1 \bigwedge (C_2 \vee \neg x_1) \bigwedge (C_2 \vee x_1 \vee \neg x_2) \bigwedge (C_2 \vee x_1 \vee x_2) \tag{2}$$

The first three clauses are independent. If we repeat the process of making the last clause independent with the previous ones until *xp* is considered; we have that *C1* $\bigwedge$ *C2* can be written as:

$$C_1 \bigwedge (C_2 \vee \neg x_1) \bigwedge (C_2 \vee x_1 \vee \neg x_2) \bigwedge \ldots \bigwedge (C_2 \vee x_1 \vee x_2 \vee \ldots \vee \neg x_p) \bigwedge (C_2 \vee x_1 \vee x_2 \vee \ldots \vee x_p) \tag{3}$$

The last clause contains all literals of C_1, so it is subsumed by C_1, and then.

$$C_1 \bigwedge C_2 \equiv C_1 \bigwedge (C_2 \vee \neg x_1) \bigwedge (C_2 \vee x_1 \vee \neg x_2) \bigwedge \ldots \bigwedge (C_2 \vee x_1 \vee x_2 \vee \ldots \vee \neg x_p) \tag{4}$$

We obtain on the right side of (4) an independent set of $p + 1$ clauses, which we denote as ind_red(C_1, C_2).

We consider a new information ϕ that is a conjunction of clauses φ_j, $j = 1,\ldots,m$ in propositional calculus. In order to review if each clause is inferred from a *KB K*, we must consider when $K \vDash \varphi$, for φ in ϕ, and for this, we must review if $C \vDash \varphi$, for any clause C in K. Then, we introduce a belief revision operator in order to review if $C \vDash \varphi$.

We will use the independent reduction between two clauses C_1 and φ (or between their respective falsifying strings) to define:

$$Ind(\varphi, C) = \begin{cases} \varphi & \text{if } \varphi \text{ and } C \text{ are independent} \\ \emptyset & \text{if } Lit(C) - Lit(\varphi) = \emptyset \\ ind_red(C, \varphi) - C & \text{in other case} \end{cases} \tag{5}$$

It is easy to redefine the operator *Ind* in terms of the falsifying strings representing *Fals*(C) and *Fals*(φ), see for example [8] for a construction based on falsifying strings. The operation *Ind*(φ,C) forms a conjunction of clauses whose falsifying assignments are exactly *Fals*(φ) – *Fals*(C).

3 A Belief Revision Operator

The paradigm best known for belief revision is the AGM paradigm [9]. Subsequently, Mendelzon and Katsuno [10] unified the different belief revision approaches to semantic, and reformulated the AGM postulates, which were called: KM postulates. Subsequently, Darwiche and Pearl [11] proposed the iterated revision, where his proposal establishes a representation based on model assumptions.

The problem of belief revision initially requires knowing whether $K \models \phi$, which involves solving a difficult problem (a Co-NP complete problem). One way to practically solve these types of problems is to introduce approximate algorithms for changing beliefs, as well as algorithmic proposals for revising finite belief bases. However, this leads to generating non-optimal solutions.

In this work, instead of computing the set of models of the involved formulas, the revision operator works on the sets of falsifying assignments of the formulas. And our belief revision's method is based on the following property.

Theorem 1. Given φ *and* C *two clauses, it holds:* $(C \bigwedge Ind(\varphi,C)) \models \varphi$.

Proof: If φ and C are independent, then $Ind(\varphi,C) = \varphi$. Therefore, $(C \bigwedge Ind(\varphi,C)) \cong (C \bigwedge \varphi)$, and in this case we have that $(C \bigwedge \varphi) \models \varphi$ by the propositional property $(p \bigwedge q) \models q$, and by the reflexivity $\varphi \models \varphi$. Otherwise, it holds that $(C \bigwedge \varphi) \cong (C \bigwedge Ind(\varphi,C))$ since $Ind(\varphi,C)$ is a set of clauses independent with C and by (2), then $(C \bigwedge Ind(\varphi,C)) \models \varphi$.

Theorem 2. Let φ *and* C *be two clauses, then* $Fals(Ind(\varphi,C)) = Fals(\varphi) - Fals(C)$.

Proof: If $Ind(\varphi,C) = \emptyset$, then $Fals(\varphi) \subseteq Fals(C)$. Therefore, $Fals(\varphi) - Fals(C) = \emptyset$. Now, let us assume that $Ind(\varphi,C) \neq \emptyset$. Let s be an assignment such that $s \in Ind(\varphi,C)$. We will show that $s \in Fals(\varphi)$ and $s \notin Fals(C)$. If $s \in Fals(Ind(\varphi,C))$, then s falsifies φ because each clause in $Ind(\varphi,C)$ has the form $(\varphi \vee R)$ where R is a disjunction (R could be empty, for example in the case $Ind(\varphi,C) = \emptyset$). If s falsifies $(\varphi \vee R)$, then s has to falsify φ, and thus $s \in Fals(\varphi)$. On the other hand, the clause $(\varphi \vee R) \in Ind(\varphi,C)$ is independent from C because of the construction of the operator Ind; therefore, $(Fals(C) \bigcap Fals(Ind(\varphi,C))) = \emptyset$. Thus, $s \notin Fals(C)$.

Given two clauses φ_i and C_j, the operator $Ind(\varphi_i,C_j)$ forms a set of strings that represents the set of falsifying assignments: $Fals(\varphi_i) - Fals(C_j)$. This is, $Ind(\varphi_i, C_j)$ determines the assignments that are in $Fals(\varphi_i)$, but which are not contained in $Fals(C_j)$.

Let φ be a clause, and $K = \bigwedge_{i=1}^{m} Ci$ be a CF. If we apply the independence's operator between each C_j and φ, we get as result S_{ij}. The union of each S_{ij}, $S_{\varphi_i} = \bigcup_{i=1}^{m} S_{ij}$ holds that $S_{\varphi_i} \subseteq Fals(\varphi)$ and $S_{\varphi_i} \not\subset Fals(K)$.

When the operator $Ind(\varphi_i, C_j)$ is applied on all $C_j \in K$, a set S of strings is obtained. Furthermore, S holds that $S \subseteq Fals(\varphi_i)$ and $S \not\subset Fals(K)$.

The sets $Ind(\varphi_i, C_j)$ allows to build a CF Fs_i such that $S_{\varphi_i} = (D_1 \bigwedge D_2 \bigwedge \ldots D_t)$, and $S_{\varphi_i} = Fals(Fs_i)$. When the new clauses Fs_i are added to K, a new KB $K'_i = K \bigwedge Fs_i$ is obtained. K'_i holds $K'_i \models \varphi_i$, meanwhile K'_i remains as a CF.

K is extended with the clauses Fs_i obtained by $Ind(\varphi_i,K)$ forming $K' = (K \bigwedge Fs_i)$. Thus K' also extends the space of falsifying assignments of K, adding the falsifying

assignments in $Fals(\varphi_i)$ that were initially not included in $Fals(K)$. In fact, $Fals(Ind(\varphi_i, K))$ and $Fals(\varphi_i)$ are disjointed, meaning that $Fals(K) \bigcap Fals(Ind(\varphi_i, K)) = \emptyset$. In other words, the set of models of K' is now a subset of the models of K, that is, $Sat(K') \subseteq Sat(K)$.

But, the application of the operator $Ind(\varphi_i, K)$ on each $\varphi_i \in \phi$, $i = 1,\ldots,k$ could not produce a minimum number of clauses. A reduction that we call *Varcom* allows to reduce the number of clauses formed by $Ind(\phi,K)$.

Hence, after obtaining the set of clauses Fs from $Ind(\varphi,K)$, we reduce its cardinality, eliminating subsumed clauses and when two clauses $Ind(\varphi_{i1}, K)$ and $Ind(\varphi_{i2}, K)$ are different only by a complementary literals, then both clauses are the same only changing its complementary literals by the symbol '*'.

In order to generate a minimum set of independent clauses as a result of $Ind(\varphi,K)$, it is crucial to sort the clauses $C_j \in K$ in ascending order according to the length.

$|S_j| = |Lit(C_j) - Lit(\varphi)|$. This is done since the number of literals in C_j, different to the literals in φ, determine the number of independent clauses to be generated. Hence, we have a strategy for reducing the number of independent clauses to be generated by each $Ind(\varphi_i, C_j)$.

This is done in order to obtain minimal set of clauses for $Ind(\varphi, K)$. Notice that all reductions maintain conjunctive forms as outcome.

Our proposal is based on the construction of a CF: $Ind(\varphi, C)$, that is necessary to fulfill $(C \bigwedge Ind(\varphi,C)) \vDash \varphi$.

The definition of the operator $Ind(\varphi, C)$ is constructive. The operator $Ind(\varphi, C)$ allows us to propose a new way to solve the problem of belief revision between conjunctive forms.

4 Multi Agent Systems

An agent is an entity enabled to perceive its environment and capable of acting in it, according to what it senses but an intelligent agent is expected not only to act as a response to predefined stimulus, but adapting to the changes in the environment while, possibly, learning from it [12]. Multi agent systems can be seen as a distributed system composed by a network of autonomous agents with its own objectives to satisfy [13].

A Multi agent system is like a society of individuals interacting with each other, exchanging information and negotiating between them to reach its own goals [14]. This leads to the need for a mechanism that allows coordination between the actions of the agents without external supervision, as agents must reach an agreement on their own [15].

Multi agent technology is applied by intelligent systems to solve the problems of analysis of complex systems and intelligent management activities. Intelligent Multi Agent Systems (MAS) based learning combine collection of information from their environment, recognition data, intelligent classification data and prediction future data, storage data, delivery data to knowledge management systems such as Decision Support System (DSS) and Management Information System (MIS) [16].

Agent based system and intelligent agents are two important issues in research area. Intelligent multi agent systems have great potentials to use in different purposes [16]

like MAS learning, MAS in development of ambient intelligence, MAS and simulation and use of MAS for Smart Cities planning and control [13].

The knowledge of each agent in a multi agent system may be different from each other and even the knowledge of one agent may be in conflict with the knowledge of another. With the purpose that a MAS can make a collective decision given a new information φ, and where each agent $A_i \in NK$ has associated an own KB K_i, we propose that each agent apply a belief revision operator of K_i on the new information φ, and according to the degree of acceptation of φ by A_i, then to make a collective decision of all MAS. We present in the following section how to make such collective decision.

5 A Network Knowledge Base

According to [17], a network knowledge base (*NKB*) is a directed graph with a labeling function for vertices and another for edges. The labels express attributes and its values. Each of the vertices is associated with an user of a social network with its own knowledge base and the edges are the relationship between users. This approach aims to predict reactions of users of a social network to the news received thru the network.

In this work, a more general approach will be used, using a MAS for deciding the acceptance of new information. Let $NK = \{A_1, A_2, \ldots, A_k\}$ be a network knowledge base formed by a set of k intelligent agents. Each agent A_i, $i = 1, \ldots, k$ has associated a local knowledge base *KB* K_i representing the beliefs currently held by the agent A_i.

Each knowledge base K_i is expressed by conjunctive forms, this is $K_i = C_{i,1} \bigwedge \ldots \bigwedge C_{i,m_i}$ where each $C_{i,j} \in K_i$ is a clause. We will omit the subindex i, when it is not necessary to identify the agent A_i.

Given a *KB* K_i, each clause $C_{ij} \in K_i$ has associated a weight W_{ij}, $j = 1, \ldots, m_i$. Each weight $0 < W_{ij} \leq 1$ indicates the degree of trust that the agent A_i has on the veracity of the clause C_{ij}. In this way, a same clause C can have different weights (different degree of trust) assigned by the different agents in the *NKB*.

The weight of the degree of belief of an agent A_i is $weight(A_i) = \sum_{j=1}^{mi} W_{ij}$. In order to normalize the knowledge of an agent in regards to the knowledge of the collective of agents, the relative weight of the agent in the network knowledge base is $rel_weight(A_i) = weight(A_i) / \sum_{j=1}^{k} weight(A_j)$.

Afterwards, we address the broad issue of understanding how new information is accepted or not in a collective way for the network of agents. For this task, we must determine first, the degree of relevance of any literal $l \in Lit(A_i)$ into the set of beliefs of the agent A_i, denoted as $rel(l, A_i)$.

We determine the importance of the literal l in the degree of belief of A_i in the following way:

$$rel(l, A_i) = \sum_{l \in C_{ij}} (W_{ij} / |C_{ij}|) \tag{6}$$

The function *rel*: $Lit(A_i)$ x $NK \rightarrow R^+$ measures the degree of relevance of a literal l into the beliefs of the agent A_i. The relative weight of a literal l in a K_i is relevant because it allows us to calculate for each agent, the relative weight of the new information. Thus, we assume that each agent adopts its own degree of belief from any new information.

The relative weight of a literal l is normalized with respect to the sum of the weights of the literals involved in the *KB* of the agent A_i. For this, we must consider first the total weight of the literals involved in K_i as: $W_l(A_i) = \sum_{l \in Lit(A_i)} rel(l, A_i)$. Therefore, the strategic value of a literal l with respect to all literals involved in a *KB* K_i of an agent A_i is: $w_rel(l, A_i) = rel(l, A_i)/W_l(A_i)$. Note that in this case, we have that $\sum_{l \in Lit(A_i)} w_rel(l, A_i) = 1$.

It is known that there are distinct strategies for collective decision-making (see e.g. [18]) depending on whether speed or accuracy is imperative and, more broadly, shed light on how intelligence is best attained. Our proposal for the collective decision-making is based on the logical value of each literal composing the new information ϕ that is shared in our MAS model.

We address the broad issue of understanding how new information ϕ is shared in the MAS model, and which is the degree of belief of each agent A_i in the MAS on ϕ. For example, we could define the strategic value of a new clause C (or new rule) to be considered by any agent $A_i \in NK$. Given a clause $C = (l_1 \vee \ldots \vee l_r)$ of literals in $Lit(A_i)$, we define the relative weight of the clause C by the agent A_i as $weight\ (C, A_i) = \sum_{l \in C} w_rel(l, A_i)$, because we assume that each agent adopts its own degree of belief from any new information. Let us consider that the new information is a CF, it is $\phi = \varphi_1 \bigwedge \ldots \bigwedge \varphi_m$.

Let S_{φ_i} be the additional information that a KB K_i requests in order to infer any $\varphi \in \phi$, this is $S_{\varphi_i} = Ind(\varphi, K_i)$. Thus, for each agent $A_i \in NK$, S_{φ_i} will be the information (given as a conjuctive formula) that is necessary to hold $(K_i \cup S_{\varphi_i}) \models \varphi$. Hence, S_{φ_i} is the information that K_i does not have, but that it is necessary to add to the beliefs of A_i in order to make A_i believes in φ.

In this way, we can weigh how much information is missing from an agent's belief set K_i, so that A_i believes in the new information φ. The information required by A_i is S_{φ_i}, which must be weighed in terms related to the importance of the literals that compose it.

We can measure the information contained in S_{φ_i}, denoted as $weight(S_{\varphi_i})$. First, we separate from S_{φ_i} the new literals in S_{φ_i} but that are not in φ. Those set of new literals are necessary to infer φ from K_i. Let us consider that $S_{\varphi_i} = (C_1 \wedge \ldots \wedge Cr)$. Let $S_{-\varphi_i} = (C'_1 \wedge \ldots \wedge C'r)$ be the set of clauses from S_{φ_i} without the literals of φ, this is, for all C_j in S_{φ_i}, $C'_j = C_j - \varphi$. And then, we can associate a weight to S_{φ_i}, denoted as $weight\big(S_{\varphi_i}\big)$, which is made according to the weight of the literals in $(S_{\varphi_i} - \varphi)$ and by the following rules:

1. If $S_{\varphi_i} = \emptyset$ then $weight\big(S_{\varphi_i}\big) = 0$.
2. If $l = (S_{\varphi_i} - \varphi), l \in Lit(A_i)$, then $weight\big(S_{\varphi_i}\big) = w_rel(l, A_i)$.
3. If $C = (S_{\varphi_i} - \varphi)$, C a clause, then weight $\big(S_{\varphi_i}\big) = \sum_{l \in C} w_rel(l, A_i)$.
4. If $S_{-\varphi_i} = (C'_1 \wedge \ldots \wedge C'r)$ is a conjunctive form, then $weight\big(S_{\varphi_i}\big) = min\{1, \sum_{C \in S_{-\varphi_i}} weight(C, A_i)$.

Then, an agent A_i, $i = 1, \ldots, k$ has a degree of belief on φ given by the formula: $belief\,(A_i, \varphi) = 1 - (weight(S_{\varphi_i}))/(weight(A_i))$. And the degree of belief of the collective *NKB* on φ is given by $belief\,(NK, \varphi) = average_{i=1}^{k}(belief\,(A_i, \varphi))$.

Given the new information ϕ to be considered by the network knowledge base, the objective is to determine if the information ϕ is accepted by the collective *NKB*, and

which is its degree of acceptance. For this, the *NKB* determines a constant value γ that is the threshold of acceptation of *NKB*. Thus, a new information ϕ will be accepted by all the agents in *NKB* if $belief(NK, \phi) > \gamma$. Otherwise, ϕ is rejected by all the agents in *NKB*.

In this case, when $average_{i=1}^{k}(belief(A_i,)) < \gamma$, then the new information ϕ is rejected by *NKB* and no change is performed on each K_i, $i = 1, \ldots, k$. Otherwise, all K_i in the NKB is updated as $K_i = (K_i \cup S_{\varphi_i})$, and each clause C_j in S_{φ_i} that is aggregated to K_i will be weighted as $Wij = weight(C_j, A_i)$.

Given the expressive power of the *NKB* model, there are plenty of domains in which this model can be useful since the integration of new information in organizations continues to be a topic of intensive research and immense practical importance [18]. For instance, in the problem of selecting the best users to be included in a marketing campaign based on social media, since users who have the greatest number of connections are not always the best candidates for being chosen as "seeds" of online marketing campaigns [7].

As one limitation of this work, is that the belief revision operator used in this model has been developed to work in propositional logic. In order to express the richness of language in social media, it will be necessary to extend the revision operator to other higher logics, as well as to consider other AI tools that capture negotiations between intelligent agents into different logics.

The proposed NKB model could be used in predicting people's reactions to information that they receive, modeling a society as a directed graph where nodes represent individuals with their respective set of beliefs, represented in the form of weighted clauses, i.e., the degree of confidence the individual has on each clause. This network could then be subjected to different news one by one, studying at each step the percentage of individuals who accept the information and predicting whether the society will collectively accept or reject the news.

This model can also be applied to small groups, such as a work team, where each member has their respective belief system, depending on their studies and experience. Each member can have different levels of confidence in their own beliefs and the confidence they have in the rest of the team members and their abilities, helping in possible collective decision-making.

For example, we have a practical application of the belief review operator in the diagnosis about the possible strain of covid that a patient may have contracted, based on the symptomatology that it shows. An immediate extension of our application is to consider a team of doctors, and based on the opinion of each one, obtain a diagnosis of the group.

Another possible application of this model is as a support tool—in combination with NLP and ML tools, among others—for detecting unfair competition. In this setting, a *NKB* could be useful in detecting information flow patterns that are typical of sock puppet campaigns.

6 Conclusions

Given a *KB* K_i associated with an intelligent agent A_i, and a new information φ, both expressed as conjunctive forms in propositional calculus, we have developed a belief revision operator on propositional logic that allows us to construct the missing information S_{φ_i} that is needed to hold $(K_i \cup S_{\varphi_i}) \models \varphi$.

A logical operator, called $Ind(\varphi, K_i)$, is proposed. $Ind(\varphi, K_i)$ finds a minimal set S_{φ_i} of clauses whose falsifying assignments cover $Fals(\varphi) - Fals(K_i)$, and therefore, $(K_i \bigwedge S_{\varphi_i}) \vDash \varphi$.

We extend this belief revision process to allow us to consider a collective decision-making by a network knowledge base *NKB*. Therefore, given new information φ, a collective decision of acceptance of φ can be made by the entire *NKB*.

In order to decide the acceptance (or not) of φ, the belief revision operator *Ind* is applied on each *KB* K_i and φ. Afterward, $S_{\varphi_i} = Ind(\varphi, K_i)$, and each one of the literals that form S_{φ_i} is weighed in order to estimate the cost that is required by A_i to accept φ.

Only in the case that φ will be accepted collectively, each one of the K_i in the network *NKB* is updated, as $K_i = (K_i \bigwedge S_{\varphi_i})$. Otherwise, there are no changes to all *KB* K_i in the *NKB*.

Given the expressive power of the *NKB* model, there are plenty of domains in which this model can be useful since the integration of new information in organizations continues to be a topic of intensive research and immense practical importance.

Acknowledgment. The authors wish to thank to the Conacyt (Consejo Nacional de Ciencias y Tecnología) – México.

Conflict of Interest. "The authors declare no conflict of interest".

References

1. Shankar, N.: Metamathematics, machines and gödel's proof. In: Cambridge Tracts in Theoretical Computer Science, No. 38. Cambridge University Press (1997)
2. Jansana, R.: Propositional Consequence Relations and Algebraic Logic. In: Zalta, E.N. (ed.) (2006)
3. Bordeaux, L., Hamadi, Y., Zhang, L.: Propositional satisfiability and constraint programming: a comparative survey. ACM Comput. Surv. **38**, 1–54 (2006)
4. Gallier, J.: SLD-resolution and logic programming (PROLOG). In: Logic for Computer Science, Foundations of Automatic Theorem Proving, pp. 410–447. University of Pennsylvania, Department of Computer and Information Science (2003)
5. De Ita, G., Marcial-Romero, R., Bellor, P., Contreras, M.: Belief revision between conjunctive normal forms. J. Intell. Fuzzy Syst. **34**(5), 3155–3164 (2018)
6. Bello, P., De Ita, G.: An algorithm to belief revision and to verify consistency of a knowledge base. IEEE Latin America Trans. **19**(11), 1867–1874 (2021)
7. Gallo, F., Simari, G., Martinez, M.V., Santos, N.A., Falappa, M.A.: Local belief dynamics in network knowledge Bases. ACM Trans. Comput. Logic **23**(1), 1–36 (2021)

8. De Ita, G., Marcial-Romero, R., Hernández, J., Pozos-Parra, P.: Using binary patterns for counting falsifying assignments of conjunctive forms. Elect. Notes in Theoretical Comput. Sci. **315**, 17–30 (2015)
9. Alchourron, C., Gardenfords, P., Makinson, D.: On the logic of theory change: partial meet contraction and revision functions. J. Symbolic Logic **50**, 510–530 (1985)
10. Katsuno, H., Mendelzon, A.O.: On the difference between updating a knowledge base and revising it. In: KR'91 Cambridge, MA, USA, pp. 387–394 (1991)
11. Darwiche, A., Pearl, J.: On the logic of iterated belief revision. Artif. Intell. **89**, 1–29 (1997)
12. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice Hall Press, Upper Saddle River, NJ, USA (2009)
13. Julian, V., Botti, V.: Multi-agent systems. In: Applied Sciences (Switzerland), vol. 9, Issue 7. MDPI AG (2019). https://doi.org/10.3390/app9071402
14. Oprea, M.: Applications of multi-agent systems (2004). https://doi.org/10.1007/1-4020-8159-6_9
15. Wooldridge, M.: An Introduction to MultiAgent Systems. John Wiley & Sons, Chichester (2002)
16. Asadi, R., Mustapha, N., Sulaiman, N.: A framework for intelligent multi agent system based neural network classification model. Int. J. Comput. Sci. Inf. Secur. **5**(1), 168–174 (2009)
17. Gallo, F.R., Simari, G.I., Martinez, M.V., Falappa, M.A.: Predicting user reactions to Twitter feed content based on personality type and social cues. Future Generation Comput. Syst. **110**, 918–930 (2020). https://doi.org/10.1016/j.future.2019.10.044
18. van Veen, D.-J., Kudesia, R.S., Heinimann, H.R.: An agent-based model of collective decision-making: how information sharing strategies scale with information overload. IEEE Trans. Comput. Soc. Syst. **7**(3), 751–767 (2020). https://doi.org/10.1109/TCSS.2020.2986161