Encoding formulas with partially constrained weights in a possibilistic-like many-sorted propositional logic

Salem Benferhat
CRIL-CNRS, Université d'Artois
rue Jean Souvraz
62307 Lens Cedex. France.
benferhat@cril.univ-artois.fr

Henri Prade
IRIT (CNRS-UPS)
118, Route de Narbonne
31062 Toulouse Cedex 04, France.
prade@irit.fr

Abstract

Possibilistic logic offers a convenient tool for handling uncertain or prioritized formulas and coping with inconsistency. Propositional logic formulas are thus associated with weights belonging to a linearly ordered scale. However, especially in case of multiple source information, only partial knowledge may be available about the relative ordering between weights of formulas. In order to cope with this problem, a two-sorted counterpart of possibilistic logic is introduced. Pieces of information are encoded as clauses where special literals refer to the weights. Constraints between weights translate into logical formulas of the corresponding sort and are gathered in a distinct auxiliary knowledge base. An inference relation, which is sound and complete with respect to preferential model semantics, enables us to draw plausible conclusions from the two knowledge bases. The inference process is characterized by using "forgetting variables" for handling the symbolic weights, and hence an inference process is obtained by means of a DNF compilation of the two knowledge bases.

1 Introduction

Information is often pervaded with uncertainty, and logics of different types have been developed for handling uncertain pieces of knowledge, some based on probability theory, others based on non-additive formalisms such as possibility theory. The lack of total certainty of a piece of information is then assessed by means of an evaluation that estimates the degree of certainty of the piece of information under the form of a precise value, or at least of a bound that constrains its value. For instance, a possibilistic logic expression of the form (p, a) encodes the constraint $N(p) \geq a$, where p is a proposition, $a \in (0, 1]$, and N is a necessity measure [Dubois *et al.*, 1994], with the intended meaning that p is certain at level at least a.

In this paper, only partial information is supposed to be available about the relative ordering between the weights of the formulas in the base. More precisely, propositional formulas are associated with symbolic weights, and a set of constraints on these weights is specified to express the relative importance of these weights. These weights may be compound symbolic expressions (e.g., as the result of formal computations) involving maximum and minimum operations.

For handling these symbolic weights, we propose an approach that uses a logical encoding of the weights. Indeed, one may think of expressing that a piece of information p is not totally certain, by stating that p is true *under some (unspecified) condition*, namely "things are not abnormal". Thus, the uncertain statement p can be written $p \vee A$, which can be read "p is true or it is abnormal (A)". Then note that if one simultaneously asserts $p \vee A$ and $\neg p \vee A'$, i.e. it is somewhat certain that p is true and it is also somewhat certain that p is false, $A \vee A'$ can be derived from the two pieces of information, which expresses that in a way or another we are in an abnormal situation. In this paper, since $p \vee A$ will be used as an encoding of the possibilistic formula (p, a), A may be a propositional formula if a is a symbolic expression. For instance, the case of multiple abnormality situations for p will be encoded by $p \vee A_1 \vee ... \vee A_n$.

This suggests a way of dealing with propositional formulas with partially constrained weights in a purely logical setting. In possibilistic logic, formulas with a weight strictly greater than the inconsistency level of the knowledge base are immune to inconsistency and can be safely used in deductive reasoning. In the proposed approach, the weights attached to the inferred formulas are handled as symbolic expressions, as well as the inconsistency level of the base. A procedure is described which enables us to determine when the available information is enough for knowing if a symbolic weight is greater than another or not.

The paper proposes a unified and general framework to represent and handle partially-constrained weighted formulas, using propositional logic. Its main contributions are:

- to encode available pieces of information Σ using a propositional knowledge base K_Σ. Each symbolic weight a is represented by a propositional expression A. Each compound expression is associated with a propositional formula, obtained by replacing maximum by a conjunction, and minimum by a disjunction. An uncertain formula (p, a) will be represented by a propositional logical formula $p \vee A$, where p and A are built using two different sets of variables: V and S. Intuitively, $p \vee A$ means that p is uncertain, and its uncertainty degree is encoded by a logical formula A.

- to encode constraints C on symbolic weights with another propositional knowledge base K_C. The inequality $a \geq b$ will be represented using material implication. The fact that a is at least as large as b, will be encoded by a propositional formula $A \to B$ (i.e., $\neg A \vee B$). Maximum and minimum operations are also encoded here using respectively conjunctions and disjunctions.
- to define an inference relation in order to draw plausible conclusions from K_Σ and K_C. This inference is sound and complete with respect to a semantics based on preferential models. It extends the possibilistic logic (where the constraints on symbolic weights induce a total preordering).
- to characterize this inference process by "forgetting" variables of V, and by "forgetting" negative literals from S. The inference process basically comes down to infer from K_Σ and K_C the strongest positive formula (that does not contain negation symbol), denoted $Linc(K_\Sigma, K_C)$, that only contains variables from S. Intuitively, $Linc(K_\Sigma, K_C)$ represents the logical counterpart of the inconsistency degree of the knowledge base.
- to use recent results on compilations (e.g., [Darwiche, 2004]) to compute plausible inferences. More precisely, the knowledge base $K_\Sigma \cup K_C$ is first compiled into DNF. These formats allow then to have a linear computation of $Linc(K_\Sigma, K_C)$.

After a brief survey of possibilistic logic in Section 2, Section 3 states the problem of reasoning from possibilistic formulas with partially constrained uncertainty weights. Section 4 provides its purely logical counterpart as a two-sorted logic and defines a sound and complete inference process from the two knowledge bases encoding respectively the pieces of uncertain information, and the constraints on the uncertainty levels. The handling of symbolic weights in the inference process is then characterized in section 5 in terms of forgetting variables, and in terms of DNF (or d-DNNF compilation) of the knowledge bases. This section also briefly considers the case of totally ordered information corresponding to standard possibilistic knowledge bases.

2 Brief background on possibilistic logic

We start with a brief refresher on possibilistic logic (for more details see [Dubois *et al.*, 1994]). A possibilistic logic formula is a pair made of a classical logic formula and a weight a expressing certainty. The weight $a \in (0, 1]$ of a formula p is interpreted as the lower bound of a necessity measure N, i.e., the possibilistic logic expression (p, a) is understood as $N(p) \geq a$. Since $N(p \wedge q) = min(N(p), N(q))$ it is always possible to put a possibilistic formula (p, a) under the form of a conjunction of clauses (p_i, a) if $p \equiv \wedge_i\, p_i$. The basic inference rule in possibilistic logic put in clausal form is the resolution rule: $(\neg p \vee q, a); (p \vee r, b) \vdash (q \vee r, min(a, b))$. Classical resolution is retrieved when all the weights are equal to 1. Let $\Sigma = \{(p_i, a_i) : i = 1, n\}$ be a knowledge base. The level of inconsistency of Σ is defined as:

$Inc(\Sigma) = max\{a : \Sigma_a \vdash \bot)\}$ (by convention $max\emptyset = 0$). where $\Sigma_a = \{p_i : (p_i, a_i) \in \Sigma,$ and $a_i \geq a\}$.

It can be shown that $Inc(\Sigma) = 0$ iff $\Sigma^* = \{p_i : (p_i, a_i) \in \Sigma\}$ is consistent in the usual sense.

Refutation can be easily extended to possibilistic logic. Proving (p, a) from Σ amounts to adding $(\neg p, 1)$, put in clausal form, to Σ, and using the above rules repeatedly until getting $\Sigma \cup \{(\neg p, 1)\} \vdash (\bot, a)$. Clearly, we are interested here in getting the empty clause $\bot$ with the greatest possible weight, i.e., $Inc(\Sigma \cup \{(\neg p, 1)\}) \geq a$. Indeed, the conclusion (p, a) is valid only if $a > Inc(\Sigma)$.

Semantic aspects of possibilistic logic, including soundness and completeness results with respect to the above syntactic inference, are presented in [Dubois *et al.*, 1994]. Semantically, a possibilistic knowledge base $\Sigma = \{(p_i, a_i) : i = 1, n\}$ is understood as a complete pre-ordering on the set of interpretations: $\omega >_\Sigma \omega'$ iff $max\{a_i : (p_i, a_i) \in \Sigma$ and $\omega' \not\models p_i\} > max\{a_j : (p_j, a_j) \in \Sigma$ and $\omega \not\models p_j\}$. Thus ω is all the less plausible as it falsifies formulas of higher degrees.

3 Possibilistic logic with symbolic weights

3.1 Representing beliefs

Let V be a set of propositional variables. Let L_V be a propositional language built from V using the propositional connectors $\wedge, \vee, \neg$. Let H be a set of symbolic weights or variables. Each symbol in H takes its value in the interval $[0, 1]$.

In the following, ordinary propositions are denoted by lower case letters $p, q, r, ...$, symbolic weights are denoted by lower case letters from the beginning of the alphabet a, b, c....

Let $\Sigma = \{(p_i, a_i) : i = 1, ..., n\}$ be a knowledge base with symbolic weights, where (p_i, a_i) expresses that p_i is a believed with a symbolic weight a_i. a_i can be a compound expression, namely a max/min expression. More precisely, max/min expressions are obtained only using the two following rules: i) $a \in H$ is a max/min expression, ii) if a and b are max/min expressions, then $max(a, b)$ and $min(a, b)$ are also max/min expressions.

Example 1 *Let p and r be two propositional symbols. Let a, b, c, d, e be five symbolic weights. In the following, we will use the following base:*

$$\Sigma = \{(p, max(a, b)), (\neg p \vee r, min(c, d)), (\neg r, e))\},$$

to illustrate the main concepts of the paper. Σ reflects that p is asserted by two sources having reliability a and b respectively, while for $\neg p \vee r$ it is unsure if the information should be considered as having reliability equal to c or to d.

3.2 Representing constraints

Constraints bearing on symbolic weights are described by a set of inequalities. A simple form of constraints is: $a_i \geq b_i$, where a_i and b_i are elements of H. These constraints $a_i \geq b_i$ restrict the possible values that a_i and b_i can take in $[0, 1]$.

More generally, the set of constraints are inequalities between max/min expressions of the form:

$C = \{a_i \geq b_i : i = 1, .., s$ and
a_i and b_i are max/min expressions$\}$.

Note that any set of constraints C can be equivalently rewritten into a canonical form:

$C = \{max(a_{i1}, ..., a_{in}) \geq min(b_{i1}, ..., b_{im}) : i = 1, .., r,$ and $a_{ij} \in H, b_{jk} \in H\}$.

This follows from the facts that i) $a \geq max(b, c)$ is equivalent to $a \geq b$ and $a \geq c$, ii) $min(b, c) \geq a$ is equivalent to $b \geq a$ and $c \geq a$, and iii) max and min are distributive.

In the following, we assume that constraints in C are in this canonical form.

Given a set of constraints C, we are interested in checking whether a given equality $a \geq b$ (a and b are max/min expressions) follows from this set of constraints.

Definition 1 *Let a and b be two max/min expressions.*

- *An assignment I is a function that assigns to each symbolic weight a degree belonging to* $(0, 1]$.
- *An assignment I is a solution of C if it satisfies each constraint of C.*
- *$a \geq b$ follows from C if each solution of C is also a solution of $a \geq b$.*

The derivation of strict inequality can be defined recursively as follows. Let a and b be two elements of H, then $a > b$ follows from C iff $a \geq b$ follows from C and $b \geq a$ does not follow from C (i.e., there is no proof for $b \geq a$). Now, let a_1, ..., a_n, b_1, ..., b_m be symbolic weights in H. Then $max(a_1, ..., a_n) > min(b_1, ..., b_m)$ is derived from C if there exist a_i and b_j such that $a_i > b_j$. Lastly, the derivation from C of strict inequality between two general max/min expressions can be defined using the fact that this derivation is equivalent to a derivation of a set strict inequalities of the canonical form $max(a_1, ..., a_n) > min(b_1, ..., b_m)$, with a_i's and b_i's as symbolic weights.

Example 2 *With the base of the above example, we will also consider the following set of constraints :*

$C = \{a \geq e, c \geq a, d \geq e\}$.

Note that since e $\geq$ a does not belong to the reflexive and transitive closure of C, we may consider using a closed world assumption that the inequality is strict. The same holds for other inequalities.

3.3 Plausible inference and semantics

Given a knowledge base with symbolic weights, $Inc(\Sigma)$ is now a max/min expression of symbolic weights and the plausible inference of p amounts to establish from C that $Inc(\Sigma \cup \{(\neg p, 1)\}) > Inc(\Sigma)$.

Here the weight 1 continues to be the top certainty level, i.e., $\forall a \in H, 1 \geq a$.

Our semantics is based on preferential models. We use the principle of best-out ordering defined in [Benferhat *et al.*, 1993], to derive from Σ a partial pre-order on the set of interpetations, denoted by $>_\Sigma$, in agreement with possibilistic logic. Let ω, and ω' be two interpretations. Then:

$\omega >_\Sigma \omega'$ iff for each (p_j, b_j) such that $\omega \not\models p_j$, there exists (q_i, a_i) such that $\omega' \not\models q_i$ and $a_i > b_j$ follows from C.

Then, a conclusion q is said to be a plausible consequence of (Σ, C) if q is true in all preferred models (w.r.t. $>_\Sigma$).

An important result is the complete and soundness result, namely :

Proposition 1 *$Inc(\Sigma \cup \{(\neg p, 1)\}) > Inc(\Sigma)$ follows from C iff p is true in all preferred models w.r.t $>_\Sigma$.*

Example 3 *Let us consider the above example, where :* $\Sigma = \{(p, max(a, b)), (\neg p \vee r, min(c, d)), (\neg r, e))\}$, *and* $C = \{a \geq e, c \geq a, d \geq e\}$.

Given Σ and C we are interested to check if r follows from Σ and C. Let us compute $Inc(\Sigma)$. By definition, we have:

$$Inc(\Sigma) = min(max(a, b), min(c, d), e). \quad (1)$$

Let us simplify this expression, with the help of the constraints in C, we have:
$Inc(\Sigma) = min(max(a, b), min(c, d), e)$
$= max(e, min(b, e))$, *hence:*

$$Inc(\Sigma) = e. \quad (2)$$

We now need to compute $Inc(\Sigma \cup (\neg r, 1))$, we get:
$Inc(\Sigma \cup (\neg r, 1)) = max(Inc(\Sigma), min(max(a, b), min(c, d)))$
$= max(e, max(min(a, c, d), min(b, c, d)))$
$= max(e, min(a, d), min(b, c, d))$
$= max(min(a, d), min(b, c, d))$.

It can be checked from C that $Inc(\Sigma \cup \{(\neg r, 1)\}) > Inc(\Sigma)$, since for instance $min(a, d) > e$.

4 Propositional logic encoding of partially-constrained weighted formulas

In standard possibilistic logic (recalled in Section 2), the weights a_i's are assumed to be known. In this section, these weights are only partially known. Propositions are then associated with symbolic weights or variables. These symbolic weights are related by a set of constraints. The following subsections describe in detail the representation of uncertain beliefs and constraints, in a propositional logic setting.

4.1 Encoding constraints

This subsection presents the encoding of constraints on the set of symbolic weights using propositional logic. In the following we associate to each symbolic weight a of the knowledge base a propositional symbol denoted by the corresponding capital letter A. We denote by S the set of propositional symbols associated with H (with $S \cap V = \emptyset$). Let L_S be the propositional language built from S using the propositional connectors $\wedge, \vee, \neg$.

One possible way to check if $a \geq b$ follows from C is to use propositional logic, and encode C as a set of clauses.

Given C, its encoding in propositional logic is immediate. Namely, a constraint $a \geq b$ is translated into $\neg A \vee B$ in the agreement with the fact that a and b are lower bounds (of a necessity measure) and thus a refers to the set of numbers $[0, 1]$ and $[a, 1] \subseteq [b, 1]$ holds iff $a \geq b$. The translation of $a \geq b$ into $\neg A \vee B$ can be read as as "if the situation is at least very abnormal (A), it is at least abnormal (B)" (indeed, the greater a, the more certain p in (p, a), and the more exceptional a situation where p is false).

To refer to the maximum (max), we use the conjunction operator ($\wedge$), namely max(a,b) will be encoded by $A \wedge B$. Indeed, the tautology $\neg(A \wedge B) \vee A$ reflects $max(a,b) \geq a$. To refer to the minimum *(min)*, we use the disjunctive operator ($\vee$), namely *min(a,b)* will be encoded using $A \vee B$. A clause $\neg A_1 \vee ... \vee \neg A_m \vee B_1 \vee ... \vee B_n$ will hence encodes a constraint $max(a_1, ..., a_m) \geq min(b_1, ..., b_n)$. More formally,

Definition 2 *Let C be a set of constraints. The propositional logic base associated with C, denoted by K_C, is defined by :* $K_C = \{\neg A_1 \vee ... \vee \neg A_m \vee B_1 \vee ... \vee B_n : max(a_1, ..., a_m) \geq min(b_1, ..., b_n) \in C\}$.

Example 4 *A total order* $C = \{a_1 \geq a_2, a_2 \geq a_3, ..., a_{n-1} \geq a_n\}$ *is encoded by :* $K_C = \{\neg A_i \vee A_{i+1} : i = 1, .., n-1\}$.

The following proposition shows that inequalities induced from C can be obtained using our propositional encoding:

Proposition 2 *Let a, and b be two symbolic weights. Then : $a \geq b$ follows from C iff $K_C \models \neg A \vee B$.*

This proposition can be easily generalized for any inequality of the form $a \geq b$ where a and b are max/min expressions, using remarks of Section 3.2. Namely any derivation of inequalities between max/min expressions, can be redefined in terms of derivations between symbolic weights. For instance, it can be checked that $a > b$ follows from C iff $K_C \models \neg A \vee B$ holds but $K_C \models \neg B \vee A$ does not hold.

We call S-positive formulas the formulas built from S by only using the conjunction and disjunction operator. For instance, $A \vee B$ is an S-positive formula, while $\neg A \vee B$ is not an S-positive formula.

4.2 Encoding uncertain information

As suggested in the introduction, the idea is to manipulate symbolic weights as formulas. Thus, a possibilistic formula (p, a) is associated with the classical clause $p \vee A$ where A means something as "the situation is abnormal". Interestingly enough, this view agrees with the qualitative representation of uncertainty in terms of lower bounds of a necessity measure used in possibilistic logic. The following definition gives the propositional logic encoding of possibilistic knowledge base:

Definition 3 *Let $\Sigma = \{(p_i, a_i)\}$ be a possibilistic knowledge base. Let A_i be a S-positive formula associated with a_i (by replacing in a_i the minimum with the disjunction, and the maximum by the conjunction). Then the propositional base associated with Σ, denoted by K_Σ, is defined by : $K_\Sigma = \{p_i \vee A_i : (p_i, a_i) \in \Sigma$ and A_i is the S-positive formula associated with $a_i\}$*

4.3 Characterizing plausible inferences

Until now, we have shown how to encode in propositional logic uncertain beliefs Σ and the constraints C on weights associated with these beliefs. This section defines the notion of plausible conclusions that can be drawn from (Σ, C). The set K_C is used at two stages: first it is used for simplifying the expression of the inconsistency degree, and then it is used to check, in the refutation, if the inconsistency degree of the augmented base increases.

Definition 4 *An S-prime formula of $K_\Sigma \cup K_C$ is an S-positive formula, denoted by $Linc(K_\Sigma)$, such that i) $K_\Sigma \cup K_C \models Linc(K_\Sigma)$, and ii) there is no ψ (not equivalent to $Linc(K_\Sigma)$) such that $K_\Sigma \cup K_C \models \psi$ and $\psi \models Linc(K_\Sigma)$.*

Up to logical equivalence, $Linc(K_\Sigma)$ is unique.

This definition allows to have a sound and complete inference relation with respect to the semantics given above. Indeed, the following proposition shows that $Linc(K_\Sigma)$ is the logical counterpart of $Inc(\Sigma)$.

Proposition 3 *Let K_Σ and K_C. Let $\delta(\Sigma)$ be a propositional formula obtained from $Inc(\Sigma)$ by replacing maximum by a conjunction, minimum by a disjunction, and the symbolic weights by their associated literals. Then $\delta(\Sigma)$ is logically equivalent to $Linc(K_\Sigma)$ given in Definition 4.*

5 Computing plausible inference using DNF formats

We propose a characterisation of plausible inference using the idea of forgetting variables (see for instance [Lang *et al.*, 2003; Darwiche and Marquis, 2004] for more details). Forgetting a variable p from K comes down to remove any reference of p in K.

Definition 5 *Let p be a propositional symbol of V. Then :* $ForgetVariable(K, p) = K_{p=\bot} \vee K_{p=\top}$

$K_{p=\bot}$ (resp. $K_{p=\top}$) is the knowledge base obtained from K by replacing p by false (resp. true). To forget a set of variables, we forget variable by variable, namely if A denotes a set of variables, then: $ForgetVariable(K, A) = ForgetVariable(ForgetVariable(K, p), A - \{p\})$.
It is also possible to only forget literals (atoms or negated atoms):

Definition 6 *Let l be a literal. Then :* $ForgetLiteral(K, l) = K_{l=\top} \vee (\neg l \wedge K)$

Some properties of ForgetVariable ([Darwiche and Marquis, 2004], [Lang *et al.*, 2003]), viewing a base as a conjunct of its formulas:

(1) $ForgetVariable(\phi \vee \psi, A) = Forgetvariable(\phi, A) \vee Fogetvariable(\psi, A)$.

(2) if ψ does not contain any varaible of A, then $ForgetVariable(\phi \wedge \psi, A) = \psi \wedge ForgetVariable(\phi, A)$

(3) if ϕ is a consistent conjunction of literals, then $ForgetVariable(\phi, A)$ consists in removing the variables in A from ϕ.

ForgetLiteral satisfies (1) and (2) which is enough for the purpose of the paper. The following shows that getting $Linc(K_\Sigma)$ is equivalent to first forget all formulas of the language, and then all negative literals of S

Proposition 4 *Let NegS be the set of negative literals in $K_\Sigma \cup K_C$. Let $F = ForgetVariable(K_\Sigma, V)$. Then $Linc(K_\Sigma)$ is equivalent to $ForgetLiteral(K_C \wedge F, NegS)$.*

This result is very important since it provides an efficient way to draw plausible conclusion from Σ and C. Indeed, forgetting a variable (resp. literal) can be achieved in a polynomial time if knowledge bases are in some formats like DNF or d-DNNF [Darwiche and Marquis, 2004]. The procedure for checking if a proposition p can be derived from Σ and C can be described as follows:

Step 1: transform Σ and C into K_Σ and K_C

Step 2: Put K_Σ into a DNF (or d-DNNF) form

Step 3.1: Forget Variables of V from K_Σ

Step 3.2: Forget Negated atoms of S from $K_C \cup K_\Sigma$. This gives $Linc(K_\Sigma)$.

Step 4: Put $K_\Sigma \cup \{\neg p\}$ into a DNF (or d-DNNF) form

Step 5.1: Forget Variables of V from $K_\Sigma \cup \{\neg p\}$

Step 5.2: Forget Negated atoms of S from $K_C \cup K_\Sigma \cup \{\neg p\}$. This gives $Linc(K_{\Sigma\cup\{\neg p\}})$.

Step 6: Use proposition 2 and results of Section 4.1 to check if p is a plausible consequence of Σ and C or not.

Forgetting a variable in a DNF amounts to forget it in each term, and forgetting it in a term amounts just to suppress the term. This clearly shows that this is polyomial in time. A similar procedure applies as well to d-DNNF format. This format known as Deterministic, Decomposable Negation Normal Form has been proposed recently [Darwiche, 2004] is a compact format, and has allowed the computation of generally intractable logical queries in time polynomial in the form size. An algorithm has been presented in [Darwiche, 2004] for compiling Conjunctive Normal Forms into d-DNNF directly. Our approach can clearly take advantage of this format as well.

Let us illustrate the above procedure with the following example:

Example 5 *Let us consider again Example 3, where we have:*$\Sigma = \{(p, max(a, b)), (\neg p \vee r, min(c, d)), (\neg r, e))\}$, *and* $C = \{a \geq e, c \geq a, d \geq e\}$. *Let* A, B, C, D, E *be the propositional symbols associated with symbolic weights* a, b, c, d, e.

Step 1 : Encoding Σ and C

The encoding of C in propositional logic gives :

$$K_C = \{\neg A \vee E, \neg C \vee A, \neg D \vee E\}$$

The encoding of Σ gives :

$K_\Sigma = \{p \vee (A \wedge B), \neg p \vee r \vee C \vee D, \neg r \vee E\}$.

Step 2 : Putting K_Σ into a DNF form

We first put K_Σ in a DNF form, which gives :

$(p \wedge C \wedge \neg r) \vee (p \wedge D \wedge \neg r) \vee (A \wedge B \wedge \neg p \wedge \neg r) \vee (A \wedge B \wedge C \wedge \neg r) \vee (A \wedge B \wedge D \wedge \neg r) \vee (p \wedge r \wedge E) \vee (p \wedge C \wedge E) \vee (p \wedge D \wedge E) \vee (A \wedge B \wedge \neg p \wedge E) \vee (A \wedge B \wedge r \wedge E) \vee (A \wedge B \wedge C \wedge E) \vee (A \wedge B \wedge D \wedge E)$

Step 3.1: Forgetting Variables of V

Now we forget the two variables p and r of V. By using properties (1)-(3); and after simplification, we get:

$$ForgetVariable(K_\Sigma, V) \equiv C \vee D \vee E \vee (A \wedge B)$$

Note that this is exactly the logical counterpart of $Inc(\Sigma)$ given by (1) in Example 3 (after remplacing, $\vee$ by minimum operation, $\wedge$ by maximum operation).

Step 3.2: Forgetting Negated atoms of S

Now, let us forget literals from $K_C \wedge ForgetVariable(K_\Sigma, V)$. We get:

$$Linc(K_\Sigma) = E.$$

Again this is exactly the logical counterpart of the expression (2) given in Example 3.

We are now interested to check if r is a plausible consequence of Σ and C.

Step 4 : Putting in DNF form $\Sigma \cup \{\neg r\}$

The DNF form associated with $\Sigma \cup \{\neg r\}$ is:

$(p \wedge C \wedge \neg r) \vee (p \wedge D \wedge \neg r) \vee (A \wedge B \wedge \neg p \wedge \neg r) \vee (A \wedge B \wedge C \wedge \neg r) \vee (A \wedge B \wedge D \wedge \neg r) \vee (p \wedge C \wedge E \wedge \neg r) \vee (p \wedge D \wedge E \wedge \neg r) \vee (A \wedge B \wedge \neg p \wedge E \wedge \neg r) \vee (A \wedge B \wedge C \wedge E \wedge \neg r) \vee (A \wedge B \wedge D \wedge E \wedge \neg r)$

Step 5.1. : Forgetting variables of V in $\Sigma \cup \{\neg r\}$

$ForgetVariable(\Sigma \cup \{\neg r\}, V) = (C \wedge \neg A) \vee (D \wedge \neg A) \vee (C \wedge E) \vee (D \wedge E) \vee (A \wedge B \wedge E)$

Step 5.2 : Forgetting Negative atoms

Forgetting negative literals of S gives :

$Linc(K_\Sigma \cup \{\neg r\}) = (A \wedge B \wedge E) \vee (C \wedge E \wedge A) \vee (D \wedge E)$

Step 6: Checking plausible inference

Using results of Section 4.1, it can be checked that r is a plausible consequence of Σ and C, as we have already seen in Example 3 that .

We finish this section by briefly discussing the case where weights associated with constraints are totally ordered. Let $\Sigma = \{(p_i, a_i) : i = 1, ..., n\}$. We assume without loss of generality that $a_1 \geq a_2 \geq ... \geq a_n$. The knowledge bases K_Σ and K_C are :

$K_\Sigma = \{p_i \vee A_i : (p_i, a_i) \in \Sigma\}$, and

$K_C = \{\neg A_i \vee A_{i+1} : i = 1, ..., n-1\}$.

Let D be the result of putting $K_\Sigma \cup K_C$ into a DNF form, and $Linc(K_\Sigma, K_C)$ be the result of forgetting variables of V and negative atoms from H. Then for totally ordered weights, it is possible to compute a compiled base (as in standard possibilistic logic), denoted by $Cons(K_\Sigma \cup K_C)$, as follows:

- If the inconsistency degree of Σ is a_j (namely $Inc(\Sigma) = a_j$), then $Cons(K_\Sigma \cup K_C)$ is logically equivalent to $ForgetVariable(\neg A_1 \wedge ... \wedge \neg A_{j-1} \wedge D, H)$
- If Σ is consistent, then $Linc(K_\Sigma, K_C)$ $Cons(K_\Sigma \cup K_C)$ is logically equivalent to $ForgetVariable(\neg A_1 \wedge ... \wedge \neg A_n \wedge D, H)$.

Note that $Cons(K_\Sigma \cup K_C)$ is under DNF form, and we have: p is a possibilistic consequence of Σ iff $Cons(K_\Sigma \cup K_C) \vdash p$.

Example 6 *Let* $\Sigma = \{(p, 0.8), (q, 0.5), (\neg p \vee \neg q, 0.4), (\neg p, 0.2)\}$. *Let A, B, C, E be propositional symbols associated respectively with the weights of the knowledge base (namely* $0.8, 0.5, 0.4, 0.2$*). The two propositional bases are:* $K_\Sigma = \{A \vee p, B \vee q, C \vee \neg p \vee \neg q, E \vee \neg p\}$, *and* $K_C = \{\neg A \vee B, \neg B \vee C, \neg C \vee E\}$.

The DNF associated with $K_\Sigma \cup K_C$ is:

$$D = (p \wedge q \wedge C \wedge E \wedge \neg A) \vee (p \wedge B \wedge C \wedge E) \vee (A \wedge B \wedge C \wedge E)$$

Let us forget variables of V, and negative atoms from H, we get :

$C \wedge E \vee B \wedge C \wedge E \vee A \wedge B \wedge C \wedge E$

which is equivalent to : $C \wedge E$ Now let us add $\neg A \wedge \neg B$ to D we get:

$D = p \wedge q \wedge C \wedge E \wedge \neg A \wedge \neg B$

Now forgetting variables of S gives : $Cons(K_\Sigma \cup K_C) = p \wedge q$, from which it can be checked that ψ is a possibilistic consequence of Σ iff $p \wedge q \vdash \psi$.

6 Related works

Using abnormality predicates in a logical setting is explicitly underlying several non-monotonic formalisms such as circumscription [McCarthy, 1980]. The idea is then to minimize abnormality and circumscribe it to a minimal number of individuals. The use of abnormality propositional literals is different here. Namely, no minimization process takes place (we are not dealing with exception-tolerant reasoning), and the symbolic weight attached to a formula p can itself be a compound formula reflecting the complex conditions under which it holds that p is true. What is proposed is more in the spirit of multiple source information, where the confidence of the information depends on the source, or on the topic, for instance.

Our representation framework for qualitative uncertainty does not require the knowledge of a complete pre-ordering of the different certainty levels attached to formulas. The uncertainty pervading a proposition may be either viewed as a precise notion, or as imprecisely stated by means of constraints that are opened to revision if new information becomes available. In both kinds of framework, one may only have a partial knowledge of the uncertainty. In the first case, uncertainty is handled under the form of absolute statements such as $g(p) = a$ (where g is an uncertainty measure and p is a proposition), or under the form of relative statements such as $p > q$ (where $>$ is a partial ordering encoding a plausibility relation expressing here that the plausibility of p is strictly greater than the one of q, as discussed in [Halpern, 1997]. Then, adding the piece of information that p entails q leads to inconsistency since if p entails q classically, it is expected that the plausibility of q is at least equal to the one of p. However, note that in possibilistic logic $N(p) \geq a$ and $N(q) \geq b$ together with the inequality $a \geq b$ is a weaker statement that $N(p) \geq N(q)$. Benferhat et al. [Benferhat *et al.*, 2004] propose a semantic approach for reasoning with partially ordered information in a possibilistic logic setting. The logical handling of formulas with unknown certainty weights together with the constraints relating these weights presented here turns to be much simpler and computationally more tractable.

7 Conclusion

The problem of reasoning with pieces of information having different confidence levels is raised by the handling of multiple source information. In case of partial information on the relative values of these levels, the problem becomes more difficult. An elegant method is proposed here for solving it, by rewriting the uncertain pieces of information in a two-sorted logic, and encoding the available information on the relative values of the certainty levels in a logical way. Putting the two bases in DNF format and using forgetting variables techniques enable us to compute the symbolic counterpart of the inconsistency level of a knowledge base in a linear way, and then to draw plausible inferences. Moreover, the representation technique that is used also provides a way for compiling a standard possibilistic knowledge base, a result analogous to the one obtained by Darwiche and Marquis [Darwiche and Marquis, 2004] for another type of weighted logic, the penalty logic, using d-DNNF format. This paper also briefly discusses the case of totally ordered weights, from which the number of extra variables (used to encode weights) and the number of extra binary clauses (used to encode the total ordering) necessary for the inference process is equal to the number of different weights used in base. The full development of a compilation of standard possibilistic knowledge bases in propositional logic is left for further research. Besides, another potential expected benefit of the approach, is a contribution to the solution of the drowning problem in possibilistic logic. Indeed, it can be checked that the symbolic inconsistency level of $K = \{p \vee A, \neg p \vee B, q \vee C\}$, is $Inc(K) = A \vee B$, where C does not appear, which should provide a way for finding out the "free" formulas in K that are not involved in any inconsistency conflict.

Acknowledgments: The authors would like to thank Pierre Marquis for his useful discussions.

References

[Benferhat *et al.*, 1993] S. Benferhat, C. Cayrol, D. Dubois, J. Lang, and H. Prade. Inconsistency management and prioritized syntax-based entailment. In *Procs. of the Thirteenth Intern. Joint Conf. on Art. Int. (IJCAI'93)*, pages 640–645, 1993.

[Benferhat *et al.*, 2004] S. Benferhat, S. Lagrue, and O. Papini. Reasoning with partially ordered information in a possibilistic framework. *Fuzzy Sets and Systems, Vol. 144, pp. 25-41*, 2004.

[Darwiche and Marquis, 2004] A. Darwiche and P. Marquis. Compiling propositional weighted bases. *Artif. Intell.*, 157(1-2):81–113, 2004.

[Darwiche, 2004] A. Darwiche. New advances in compiling cnf into decomposable negation normal form. In *Procs. of Europ. Conf. on Art. Intell. (ECAI2000)*, pages 328–332, 2004.

[Dubois *et al.*, 1994] D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 439–513. Oxford University Press, 1994.

[Halpern, 1997] J. Y. Halpern. Defining relative likelihood in partially-ordered preferential structures. *J. of AI Research*, 7:1–24, 1997.

[Lang *et al.*, 2003] J. Lang, P. Liberatore, and P. Marquis. Propositional independence - formula-variable independence and forgetting. *J. of Art. Intell. Research*, 18:391–443, 2003.

[McCarthy, 1980] J. McCarthy. Circumscription : A form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.