Compiling possibilistic knowledge bases

Salem BENFERHAT [1] and **Henri Prade** [2]

Abstract. Possibilistic knowledge bases gather propositional formulas associated with degrees belonging to a linearly ordered scale. These degrees reflect certainty or priority, depending if the formulas encode pieces of beliefs or goals to be pursued. Possibilistic logic provides a simple format that turns to be useful for handling qualitative uncertainty, exceptions or preferences. The main result of the paper provides a way for compiling a possibilistic knowledge base in order to be able to process inference from it in polynomial time. The procedure is based on a symbolic treatment of the degrees under the form of sorted literals and on the idea of forgetting variables. The number of sorted literals that are added corresponds exactly to the number of priority levels existing in the base, and the number of binary clauses added in the compilation is also equal to this number of levels. The resulting extra compilation cost is very low.

1 Introduction

Possibilistic knowledge bases [Dubois *et al.*1994] offer a representation framework for the qualitative handling of uncertainty or for the encoding of prioritized formulas expressing preferences. A key feature of possibilistic logic is its capability to deal with inconsistent knowledge bases. Indeed a possibilistic knowledge base is associated with a level of inconsistency, and the set of formulas whose associated level is strictly greater than this inconsistency level are safe from inconsistency, while the formulas with a level smaller or equal are ignored in the inference process. Adding new pieces of knowledge to the base may increase this inconsistency level, which provides a mechanism for encoding nonmonotonic reasoning [Benferhat *et al.*1999]. Clearly, the computation of the inconsistency level in possibilistic logic inference, as well as for related consequence relations that are able to also take advantage of some of the ignored formulas (drawn in inconsistencies) [Benferhat *et al.*1999], is a key issue.

Two related entailment relations can be defined from a possibilistic knowledge base. The first one yields the classical consequences of the subpart of the base made of the formulas whose certainty levels are strictly above the inconsistency level of the base, the second one yields the same consequences together with an associated certainty level (which is the greatest possible one compatible with the information in the base).

The computational complexity of possibilistic logic is the one of classical logic multiplied by the logarithm of the number of distinct levels used in the base [Lang2000]. As in classical logic, the compilation of possibilistic knowledge bases is a key issue in order to be able to process inference in a polynomial time. The method proposed in this paper takes advantage of ideas recently introduced in [Benferhat and Prade2005] for handling an extension of possibilistic logic allowing for the symbolic handling of partial preorders between the levels. However, the case of possibilistic logic, where there is a complete preorder between the levels, requires a particular adaptation of these ideas. A key issue not considered in [Benferhat and Prade2005] is then to compute to what level a formula can be inferred. Possibilistic logic levels are encoded by means of propositional symbols. Then the possibilistic logic base we start with is turned into a new base where on the one hand the original clauses are augmented with this encoding of their levels, and where on the other hand as many binary clauses as there are certainty levels are added to the base, thus for a very small extra computational cost. This rewriting makes possible the design of an efficient original algorithm for computing the result of a possibilistic inference with its associated degree in polynomial time.

The paper is organized as follows. Section 2 provides a short background on possibilistic logic, while Section 3 introduces the symbolic encoding of levels. Section 4 presents the main results for computing of plausible inference, using DNF forms and the forgetting variable technique. Section 5 describes the general procedure for computing compiled possibilistic bases. Sections 6 and 7 show that the computation of possibilistic entailment with the level associated to the consequent can be achieved in polynomial time, and provide the algorithms for doing it.

2 Brief background on possibilistic logic

We start with a brief refresher on possibilistic logic (for more details see [Dubois *et al.*1994]). Let V be a set of propositional variables. Let L_V be a propositional language built from V using the propositional connectors $\wedge, \vee, \neg$. A possibilistic logic formula is a pair made of a classical logic formula and a degree a expressing certainty or priority. The degree $a \in (0, 1]$ 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) where $p \equiv \wedge_i\ p_i$. In the following, ordinary propositions are denoted by lower case letters $p, q, r, \ldots$, numerical degrees (belonging to the interval [0,1]) are denoted by lower case letters from the beginning of the alphabet $a, b, c \ldots$. Anyway since the degrees play an ordinal role only, they could be replaced by the values in any linearly ordered scale.

The basic inference rule in possibilistic logic put in clausal form

[1] CRIL-CNRS, Université d'Artois, rue Jean Souvraz 62307 Lens Cedex. France. email: benferhat@cril.univ-artois.fr

[2] IRIT (CNRS-UPS), 118, Route de Narbonne 31062 Toulouse Cedex 04, France. email: prade@irit.fr

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 degrees are equal to 1.

Definition 1 *Let $\Sigma = \{(p_i, a_i) : i = 1, n\}$ be a possibilistic 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 rule 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 degree or level, i.e., $Inc(\Sigma \cup \{(\neg p, 1)\}) \geq a$. Indeed, the conclusion (p, a) is valid only if $a > Inc(\Sigma)$. More formally, the following defines the possibilistic inference:

Definition 2 *Let Σ be a possibilistic knowledge base, and p be a propositional formula. Then :*

- *p is a possibilistic consequence of Σ, denoted by $\Sigma \models_\pi p$, iff $Inc(\Sigma \cup \{(\neg p, 1)\}) > Inc(\Sigma)$.*
- *p is a possibilistic consequence of Σ to degree a, also denoted by $\Sigma \models_\pi (p, a)$, iff $Inc(\Sigma \cup \{\neg p\}) > Inc(\Sigma)$ and $a = Inc(\Sigma \cup \{\neg p\})$.*

Thus inferring from Σ inconsistent is equivalent to infer from the consistent subbase $\{p_i : (p_i, a_i) \in \Sigma,$ and $a_i > a\}$. The following illustrative example is used in the whole paper.

Example 1 *Let $\Sigma = \{(\neg q \vee r, .8), (q, .6), (e, .6), (\neg r, .5), (q, .3)\}$. We have: $Inc(\Sigma) = .5$. Now, let us check if r is a possibilistic consequence of Σ and to what degree. It is enough to compute:*

$$Inc(\Sigma \cup \{\neg r\}) = .6 > Inc(\Sigma).$$

Hence, r is a possibilistic consequence of Σ and its degree of inference is equal to .6.

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 Symbolic encoding of possibilistic knowledge

Let $\Sigma = \{(p_i, a_i) : i = 1, \dots, n\}$ be a standard possibilistic knowledge base. Let $H = \{a_1, \dots, a_n\}$ be the set of numerical degrees or levels in Σ. Each symbol in H takes its value in the interval [0, 1]. We assume without loss of generality that $a_1 \geq a_2 \geq \dots \geq a_n$.

The first step in the symbolic encoding of a possibilistic knowledge base is to associate to each numerical degree a_i in the knowledge base a propositional symbol denoted by the corresponding capital letter A_i. 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 $\wedge, \vee, \neg$.

We also need to encode the ordering relation between degrees. A constraint $a \geq b$ is translated into $\neg A \vee B$ in agreement with the fact that a and b are lower bounds (of a necessity measure) and thus a refers to the set of numbers $[a, 1[$ 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).

If two formulas have the same degree a, they will be associated with the same symbol A. It agrees with the equivalence $\{(p, a), (q, a)\}$ and $\{(p \wedge q, a)\}$ in possibilistic logic.

Definition 3 *A total order $a_1 \geq a_2, a_2 \geq a_3, \dots, a_{n-1} \geq a_n$ between numerical degrees present in Σ is encoded by the following set of (n-1) binary clauses: $\{\neg A_i \vee A_{i+1} : i = 1, \dots, n-1\}$.*

Completely certain formulas, such as tautologies, are supposed to have a level equal to $a_0 = 1$. In fact, the corresponding literal A_0 will be equivalent to $\bot$. Then $\forall i \geq 1, \neg A_0 \vee A_i$ holds. As suggested in the introduction, the idea is to manipulate numerical degrees as formulas [Benferhat and Prade2005]. Thus, a possibilistic formula (p, a) is associated with the classical clause $p \vee A$ where A means something as "the situation is A-abnormal". Thus $p \vee A$ means p is true or the situation is abnormal. Note that $p \vee A_0 \equiv p$.

The following definition gives the propositional logic encoding of possibilistic knowledge base:

Definition 4 *Let $\Sigma = \{(p_i, a_i) : i = 1, n\}$ be a possibilistic knowledge base. Let A_i be a propositional symbol associated with a_i. 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 associated with a_i*$\}$
$\cup \{\neg A_i \vee A_{i+1} : i = 1, \dots, n-1\}$.

As can be seen from definition 4, the size of K_Σ is exactly the one of Σ increased by n - 1 binary clauses, if there are n uncertainty levels used. So the cost of this rewriting is very low.

On our running example, this rewriting gives:

Example 2 *Let $\Sigma = \{(\neg q \vee r, .8), (q, .6), (e, .6), (\neg r, .5), (q, .3)\}$.*

Let A, B, C, D be four propositional (representing symbolic degrees) associated respectively with the four degrees present in the possibilistic knowledge base, namely : .8, .6, .5, .3.

Using the above definition, the propositional knowledge base assocaited with the knowledge base Σ is :
$K_\Sigma = \{A \vee \neg q \vee r, B \vee q, B \vee e, C \vee \neg r, D \vee q, \neg A \vee B, \neg B \vee C, \neg C \vee D\}$.
The three last binary clauses are used to encode the facts that the degree associated with A is greater than B, the one of B is greater than the one of C, and the one of C is greater than the one of D.

It is easy to check that the resolution rule now reads $\neg p \vee q \vee A, p \vee r \vee B \vdash q \vee r \vee A \vee B$, and $min(a, b)$ translates into $A \vee B$ with $A \vee B \equiv A$ if $b \geq a = min(a, b)$ (and indeed $\neg B \vee A, A \vee B \vdash A$).

4 A refresher on forgetting variables

This section recalls the notion of forgetting variables, which are very useful for characterizing possibilistic inferences (see for instance [Lang *et al.*2003, Darwiche and Marquis2004] for more details on forgetting variables). Forgetting a variable p from a set of propositional formulas 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 X denotes a set of variables, then: $ForgetVariable(K,X) = ForgetVariable(ForgetVariable(K,p), X - \{p\})$.
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 Marquis2004], [Lang *et al.*2003]), viewing *a base as a conjunct of its formulas* are:

(1) $ForgetVariable(p \vee q, X) = ForgetVariable(p,X) \vee FogetVariable(q,X)$.
(2) if q does not contain any variable of X, then $ForgetVariable(p \wedge q, X) = q \wedge ForgetVariable(p,X)$

ForgetLiteral satisfies (1) and (2), which is enough for the purpose of the paper. The computation of forgetting variables can be efficiently achieved if a propositional knowledge base is in a form such as DNF (disjunctive normal form) or d-DNNF forms. Indeed, 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 polynomial in time. A similar procedure applies as well to d-DNNF format. This format known as Deterministic, Decomposable Negation Normal Form, which has been proposed recently [Darwiche2004], 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 [Darwiche2004] for compiling Conjunctive Normal Forms into d-DNNF directly. Our approach can clearly take advantage of this format as well.

In the following, we denote $DNF(K)$ the result of putting K under DNF form. It is represented as a unique formula.

5 Characterising plausible inferences

This section proposes a characterisation of plausible inference using forgetting variables. We first provide a symbolic computation of the inconsistency degree of a knowledge base, then we present some propositions that characterize the compiled possibilistic knowledge base.

5.1 A symbolic computation of inconsistency degrees

This section presents the characterization of the inconsistency degree of Σ using forgetting variables on K_Σ. We first start with the case where Σ is consistent. This is stated by the following proposition:

Proposition 1 *Let Σ be a possibilistic knowledge base, and K_Σ be the propositional knowledge base associated with Σ using definition 4. Let NegS be the set of negative literals in K_Σ issued from S. Let $F = ForgetVariable(K_\Sigma, V)$. Then Σ is consistent if and only if $ForgetLiteral(F, NegS)$ is a tautology.*

Assume that Σ is consistent. The idea of the proof is first to note that $\{A_1 \vee p_1, \ldots, A_n \vee p_n\}$ is equivalent to :

$$\bigvee_{I \subseteq \{1,\ldots,n\}} (\wedge_{i \in I} A_i)(\wedge_{j \in complement(I)} p_j),$$

where complement(I) is the set of elements in $\{1,\ldots,n\}$ that are not in I.

In particular, this equivalent form of $\{A_1 \vee p_1, \ldots, A_n \vee p_n\}$ contains the term $p_1 \wedge \ldots \wedge p_n$. This also means that the equivalent form of Σ contains the term $\neg A_1 \wedge \ldots \wedge \neg A_{n-1} \wedge p_1 \wedge \ldots \wedge p_n$, which is consistent. Hence forgetting V and negated literals from this term leads to a tautology. Hence forgetting forgetting V and negated literals from K_Σ also leads to a tautology. The converse can also be shown in a similar way.

Now, let us analyse the case where Σ is inconsistent. The following proposition provides the characterisation:

Proposition 2 *Let Σ be a possibilistic knowledge base, and K_Σ be a propositional knowledge base associated with Σ using definition 4. Let NegS be the set of negative literals in K_Σ issued from S. Let $F = ForgetVariable(K_\Sigma, V)$. Then Σ is inconsistent to a degree a_i if and only if $ForgetLiteral(F, NegS)$ is equivalent to $A_i \wedge \ldots \wedge A_n$.*

The idea of the proof is the following: if Σ is inconsistent to a degree a_i, then this means that $\{p_1, \ldots, p_i\}$ is inconsistent. Hence, $\{A_1 \vee p_1, \ldots, A_i \vee p_i\}$ implies $\{A_1 \vee \ldots \vee A_i\}$. From $\{A_1 \vee \ldots \vee A_i\}$ and $\{\neg A_1 \vee A_2, \ldots, \neg A_{i-1} \vee A_i\}$ we conclude A_i. And from A_i and $\{\neg A_i \vee A_{i+1}, \ldots, \neg A_{n-1} \vee A_n\}$ we conclude $A_i \wedge \ldots \wedge A_n$. Therefore, all formulas of the form $A_j \vee p_j$ for $j \geq i$ can be removed from K_Σ. And K_Σ is equivalent to :
$\{\neg A_1 \vee A_2, \ldots, \neg A_{i-2} \vee A_{i-1}, A_1 \vee p_1, \ldots, A_{i-1} \vee p_{i-1}, A_i, \ldots, A_n\}$.
This knowledge base is consistent. Hence, forgetting variables of V and negated literals of S leads to $A_i \ldots A_n$.

Example 3 *Let us consider again our example, where we have :*
$\Sigma = \{(\neg q \vee r, .8), (q, .6), (e, .6), (\neg r, .5), (q, .3)\}$, *and,*
$K_\Sigma = \{A \vee \neg q \vee r, B \vee q, B \vee e, C \vee \neg r, D \vee q, \neg A \vee B, \neg B \vee C, \neg C \vee D\}$.

We already showed that Σ is inconsistent and its inconsistency degree is equal to .5. Let us show that Forgetting variables of V and negative literals of S in K_Σ is equivalent to $C \wedge D$. Remember that the symbols A, B, C, D are associated respectively with the degrees .8, .6, .5, .3.

Note that $A \vee \neg q \vee r, B \vee q$ implies $A \vee r \vee B$. Moreover, $A \vee r \vee B$ and $C \vee \neg r$ implies $A \vee B \vee C$. Lastly, $A \vee B \vee C$ and $\{\neg A \vee B, \neg B \vee C\}$ gives C. Again : C and $\neg C \vee D$ gives D.

Therefore, K_Σ is equivalent to :
$K_\Sigma \equiv \{A \vee \neg q \vee r, B \vee q, B \vee e, \neg A \vee B, C, D\}$.
Let us compute the DNF form of K_Σ, we get:

$DNF(K_\Sigma) \equiv (\neg A \wedge r \wedge C \wedge D \wedge q \wedge e) \vee (A \wedge B \wedge C \wedge D) \vee (\neg q \wedge B \wedge C \wedge D) \vee (r \wedge B \wedge C \wedge D)$

Let us now forget symbols of V but also negative literals of S.

- *Forgetting $\{r\}$ gives :*
  ForgetVariable $(K_\Sigma, \{r\})$
  $\equiv (\neg A \wedge C \wedge D \wedge q \wedge e) \vee (B \wedge C \wedge D)$.
- *Forgetting $\{r, q\}$ gives :*
  ForgetVariable $(K_\Sigma, \{r, q\}) \equiv (\neg A \wedge C \wedge D \wedge e) \vee (B \wedge C \wedge D)$.
- *Forgetting $\{r, q, e\}$ gives :*
  ForgetVariable $(K_\Sigma, \{r, q, e\}) \equiv (\neg A \wedge C \wedge D) \vee (B \wedge C \wedge D)$.
- *Lastly, forgetting negative literals of S gives :*
  ForgetLiteral(ForgetVariable $(K_\Sigma, \{r, q, e\}), \{\neg A, \neg B, \neg C\})$
  $\equiv (C \wedge D) \vee (B \wedge C \wedge D) \equiv C \wedge D$.

Hence, we get the expected result.

5.2 Characterizing compiled possibilistic knowledge bases

This section characterises the compiled possibilistic knowledge bases. More precisely, given a possibilistic knowledge base, we are interested in characterizing a propositional knowledge base, denoted by $Compiled(\Sigma)$ such that:

$$\Sigma \models_\pi p \text{ iff } Compiled(\Sigma) \vdash p.$$

Of course, we require that the inference from $Compiled(\Sigma)$ should be efficient (polynomial).

Again we start with the case where Σ is consistent. This is stated by the following proposition.

Proposition 3 *Let Σ be a consistent possibilistic knowledge base, and K_Σ be a propositional knowledge base associated with Σ using definition 4. Then: $Compiled(K_\Sigma) =$ $ForgetVariable(\neg A_1 \wedge \ldots \wedge \neg A_n \wedge DNF(K_\Sigma), S)$.*

The idea of the proof is quite immediate. Note first that $DNF(K_\Sigma)$ contains a term $\neg A_1 \wedge \ldots \wedge \neg A_n \wedge p_1 \wedge \ldots \wedge p_n$. All other terms contain at least a positive symbol of S. Hence, adding $\neg A_1 \wedge \ldots \wedge \neg A_n$ to $DNF(K_\Sigma)$ is equivalent to the term :
$\neg A_1 \wedge \ldots \wedge \neg A_n \wedge p_1 \wedge \ldots \wedge p_n$,
since all other terms will become inconsistent.

The following proposition gives the case where Σ is inconsistent.

Proposition 4 *Let Σ be an inconsistent possibilistic knowledge base, and K_Σ be a propositional knowledge base associated with Σ using definition 2. Assume that Σ is consistent to a degree a_i. Then: $Compiled(K_\Sigma) =$ $ForgetVariable(\neg A_1 \wedge \ldots \wedge \neg A_{i-1} \wedge DNF(K_\Sigma), S)$.*

Example 4 *Let us consider again our example.*
We recall that :

$DNF(K_\Sigma) \equiv (\neg A \wedge r \wedge C \wedge D \wedge q \wedge e) \vee (A \wedge B \wedge C \wedge D)$ $\vee (\neg q \wedge B \wedge C \wedge D) \vee (r \wedge B \wedge C \wedge D)$

Let us compute
$Compiled(K_\Sigma) \equiv ForgetVariable(\neg A \wedge \neg B \wedge DNF(K_\Sigma), S)$.

We have $\neg A \wedge \neg B \wedge DNF(K_\Sigma)$ is equivalent to :
$(\neg A \wedge \neg B \wedge r \wedge C \wedge D \wedge q \wedge e)$

Hence, forgetting variables of S gives :
$Compiled(K_\Sigma) \equiv r \wedge q \wedge e$.

6 Procedure for computing compiled possibilistic base

The above propositions are very important since they provide an efficient way to draw possibilistic conclusions from Σ. The procedure for checking if a proposition p can be derived from Σ can be described as follows:

The following proposition shows that the compiled knowledge base obtained at step 6 allows us to recover all possibilistic consequences of Σ.

Procedure 1: Computation of the compiled possibilistic base

```
Data: A possibilistic knowledge base $\Sigma$.
Result: A compiled base $Compiled(\Sigma)$
begin
    Step 1: Transform $\Sigma$ into $K_\Sigma$
    Step 2: Put $K_\Sigma$ into a DNF (or d-DNNF) form
    Step 3 Forget variables of $V$ from $K_\Sigma$
    Step 4 Forget negated atoms of $S$ from $K_\Sigma$.
    Step 5 Compute inconsistency degree $A_i$
    Step 6
      if $\Sigma$ is consistent then
        $Compiled(\Sigma) =$
        $ForgetVariable(\neg A_1 \wedge \ldots \wedge \neg A_n \wedge DNF(K_\Sigma), S)$
      else
        /* $\Sigma$ is inconsistent to a degree $a_i$
        $Compiled(\Sigma) =$
        $ForgetVariable(\neg A_1 \wedge \ldots \wedge \neg A_{i-1} \wedge DNF(K_\Sigma), S)$
end
```

Proposition 5 *Let $Compiled(\Sigma)$ be the propositional base obtained at end of step 6 of the above algorithm. p is a possibilistic consequence of Σ iff $Compiled(\Sigma) \vdash p$.*

The proof is immediate using Propositions 1-4.

Example 5 *Let us consider again our example. From previous computations we have:*
$Compiled(K_\Sigma) \equiv r \wedge q \wedge e$

We can check that for all propositional formula p, p is a possibilistic consequence of Σ if and only if p is classical consequence of $compiled(K_\Sigma)$, with the advantage that $compiled(K_\Sigma)$ is in DNF form, hence the inference is polynomial. For instance, we have already shown (example 1) that r is a possibilistic consequence of Σ. This obviously follows from $compiled(K_\Sigma)$.

Proposition 6 *Steps 3–6 of the above algorithm are achieved in a polynomial time, with respect to the number of terms (or conjuncts) in the compiled base (which is in DNF form).*

Indeed, steps 3-4 can be obtained in a linear time, since the knwoledge bases are in DNF form from which forgetting operations are polynomials[Darwiche and Marquis2004]. Step 4 which concerns the computation of inconsistency degree can also be done in a polynomial time thanks to Propositions 1 and 2. Indeed, it is enough to check if $A_1 \wedge \ldots \wedge A_n$ is equivalent $ForgetLiteral(F, NegS)$ for $i = 1, \ldots, n$ with $F = ForgetVariable(K_\Sigma, V)$.

7 Computing weighted consequences

This section proposes an algorithm to compute the degree of inference associated with a propositional formula p, once Procedure 1 has been used for compiling Σ and establishing p as a consequence of Σ. First, we need the following result:

Proposition 7 *Let p be a propositional formula that is a consequence of $Compiled(K_\Sigma)$. Then: $\Sigma \models_\pi (p_i, a_i)$ iff*

- $ForgetVariable(\neg A_1 \wedge \ldots \wedge \neg \mathbf{A}_i \wedge \mathbf{A}_{i+1} \ldots \wedge \mathbf{A}_n \wedge DNF(K_\Sigma), S) \vdash p_i$
- $\forall j = 1, \ldots, i-1,$ *:* $ForgetVariable(\neg A_1 \wedge \ldots \wedge \neg \mathbf{A}_j \wedge \mathbf{A}_{i+1} \ldots \wedge \mathbf{A}_n \wedge DNF(K_\Sigma), S) \nvdash p_i$

The idea of the proof is that when we add a positive literal A_i then all propositional formulas of levels $a_k \leq a_i$ are ignored. Indeed, we have A_i and $\{\neg A_i \vee A_{i+1}, \ldots, \neg A_{n-1} \vee A_n\}$ implies $\{A_{i+1}, \ldots, A_n\}$. Hence the set of formulas $\{A_{i+1} \vee p_{i+1}, \ldots, A_n \vee p_n\}$ will be subsumed by $\{A_{i+1}, \ldots, A_n\}$.

Given this proposition, the procedure for computing the degree associated with a possibilistic consequence is rather straightforward. It works iteratively, thanks to Proposition 7, since the level of p corresponds to the first A_i such that p can be proved from formulas with levels greater than a_i. So at the beginning, except those with level 1 (encoded by A_0) such as tautologies, are ignored. Then, adding $\neg A_i$ makes formulas in the corresponding stratum active in the inference process. Clearly, the procedure remains polynomial.

Procedure 2: Computing the symbolic weight associated with conclusions

```
Data: $F = DNF(K_\Sigma)$
      A propositional formula $p$ which is consequence of $Compiled(\Sigma)$
Result: A symbolic degree associated with $p$
begin
    $k \leftarrow 0$ ;
    X $\leftarrow \{A_1, \ldots, A_n\}$ ;
    while $ForgetVariable(X \cup F, S) \nvdash p$ do
        $k \leftarrow k + 1$;
        $X \leftarrow X - \{A_k\}$;
        $X \leftarrow X \cup \{\neg A_k\}$;
    return $A_k$
end
```

Example 6 *Let us consider again our example. We already checked that r is a possibilistic consequence of Σ to a degree .6. We also checked that $Compiled(K_\Sigma) \vdash r$. Let us apply the above procecdure to get the symbolic degree associated with r.*

We recall that :
$F = DNF(K_\Sigma) \equiv (\neg A \wedge r \wedge C \wedge D \wedge q \wedge e)(A \wedge B \wedge C \wedge D) \vee (\neg q \wedge B \wedge C \wedge D) \vee (r \wedge B \wedge C \wedge D)$.

- *At the beginning, we have k=0 and X=$\{A, B, C, D\}$. Then we can check that ForgetVariable $(X \cup F, S)$ is equivalent to a tautology.*
- *At the next iteration, we have k=1 and X=$\{\neg A, B, C, D\}$. Then $X \cup F$ is equivalent to:*
  $(\neg A \wedge r \wedge C \wedge B \wedge D \wedge q \wedge e)$
  $\vee (\neg A \wedge \neg q \wedge B \wedge C \wedge D) \vee (\neg A \wedge r \wedge B \wedge C \wedge D)$

  Then $ForgetVariable(X \cup F, S)$ is equivalent to a: $\neg q \vee r$ from which r cannot be derived.
- *At the second iteration, we have k=1 and X=$\{\neg A, \neg B, C, D\}$. Then $X \cup F$ is equivalent to:*
  $(\neg A \wedge \neg B \wedge r \wedge C \wedge D \wedge q \wedge e)$
  Then $ForgetVariable(X \cup F, S)$ is equivalent to $q \wedge r$ from which r can be derived. Hence, the symbolic degree associated with r is B, which is indeed associated with the degree 0.6.

Proposition 8 *The above algorithm, which computes the degree associated with a possibilistic conclusion, is achieved in a polynomial time, with respect to the number of terms (or conjuncts) in the compiled base (which is in DNF form).*

8 Conclusion

The paper has proposed an approach for handling possibilistic inference (including the computation of the formula levels) in polynomial time thanks to a compilation procedure that takes advantage of a propositional treatment of formula levels, and to the use of a forgetting variable procedure. Note also that the proposed compilation procedure cannot be derived from the one used by [Darwiche and Marquis2004] for penalty logic, which relies on summations and is quantitative in nature, while possibilistic logic is qualitative and uses max and min operations. Moreover, the results are also relevant for processing other inconsistency-tolerant inferences that extend possibilistic inference and are also based on the computation of inconsistency levels [Benferhat *et al.*1999].

A line for further research is the handling of hypothetical reasoning in this framework. Indeed, by moving literals in the level slots, namely the possibilistic formula $(\neg p \vee q, a)$ is equivalent to $(q, min(P, a))$ where P = 1 or 0 depending if p is assumed true or false, one can develop a possibilistic counterpart of ATMS [de Kleer1986]. It would be interesting to combine the ideas presented here, the handling of more general symbolic levels (as in [Benferhat and Prade2005]), and consequence-finding algorithms (e.g. [Inoue1991]) in that perpective.

Acknowledgments: This work is supported by the national project ANR Blanc MICRAC (Modèles informatiques et cognitifs du raisonnement causal).

REFERENCES

[Benferhat and Prade2005] S. Benferhat and H. Prade. Encoding formulas with partially constrained weights in possibilistic-like many-sorted propositional logic. In *Proceedings of 19th International Joint Conference on Artifi cial Intelligence (IJCAI'05)*, pages 1281–1286, 2005.

[Benferhat *et al.*1999] Salem Benferhat, Didier Dubois, and Henri Prade. An overview of inconsistency-tolerant inferences in prioritized knowledge bases. In *Fuzzy Sets, Logic and Reasoning about Knowledge. D. Dubois, H. Prade, E.P. Klement (Eds.)*, pages 395–417, 1999.

[Darwiche and Marquis2004] A. Darwiche and P. Marquis. Compiling propositional weighted bases. *Artif. Intell.*, 157(1-2):81–113, 2004.

[Darwiche2004] A. Darwiche. New advances in compiling cnf into decomposable negation normal form. In *Procs. of Europ. Conf. on Art. Intell. (ECAI2004)*, pages 328–332, 2004.

[de Kleer1986] Johan de Kleer. An assumption-based TMS and extending the ATMS. *Artifi cial Intelligence*, 28(2):127–162 and 163–196, 1986.

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

[Inoue1991] K. Inoue. onsequence-fi nding based on ordered linear resolution. In *12th Inter. JOint Conf. on artifi cial Intelligence (IJCAI'91), Sydney*, pages 158–164, 1991.

[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.

[Lang2000] Jérôme Lang. Possibilistic logic: complexity and algorithms. In D. M. Gabbay and P. Smets, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 5: Algorithms for Uncertainty and Defeasible Reasoning, pages 179–220. Kluwer Academic, Dordrecht, 2000.