

On the Compilation of Possibilistic Default Theories

Salem Benferhat

Université Lille-Nord de France,
Artois, F-62307 Lens
CRIL, F-62307 Lens
CNRS UMR 8188, F-62307 Lens
benferhat@cril.univ-artois.fr

Safa Yahi

Université Lille-Nord de France,
Artois, F-62307 Lens
CRIL, F-62307 Lens
CNRS UMR 8188, F-62307 Lens
yahi@cril.univ-artois.fr

Habiba Drias

Institut National d'Informatique
BP 68M Oued Smar
16309 Algiers, Algeria
h.drias@ini.dz

Abstract

Handling exceptions represents one of the most important problems in Artificial Intelligence. Several approaches have been proposed for reasoning on default theories. This paper focuses on a possibilistic approach, and more precisely on the MSP-entailment from default theories which is equivalent to system P augmented by rational monotony. In order to make this entailment tractable from a computational point of view, we propose here a compilation of default theories with respect to a target compilation language. This allows us to provide polynomial algorithms to derive efficiently the MSP-conclusions of a compiled default theory. Moreover, the proposed compilation is qualified to be flexible since it efficiently takes advantage of any classical compiler and generally provides a low recompilation cost when updating a compiled default theory.

Introduction

It is well known that one of the major purposes of nonmonotonic reasoning is to cope with the presence of exceptions in knowledge base systems. Some emphasis has been put on the application of nonmonotonic reasoning techniques to practical problems. For instance, in (Morgenstern 1997) several potential domains of applications, like medical reasoning, legal reasoning and reasoning in business organizations, have been identified. The scarcity of nonmonotonic reasoning applications to industry may be mainly due to the lack of tractable algorithms for performing plausible reasoning and also to the esoteric reputation of nonmonotonic reasoning literature which has become too theoretically-oriented.

A default theory Υ is given by a pair (Δ, W) where Δ is a set of default rules having exceptions or conditional assertions and W is the set of strict or hard rules that do not admit exceptions.

This paper focuses on the possibilistic handling of default theories proposed in (Benferhat, Dubois, & Prade 1998) that consists to view each conditional assertion $\alpha \rightarrow \beta$ as a constraint expressing that the situation where α and β is true has a greater possibility than the one where α and $\neg\beta$ is true which is expressed by $\Pi(\alpha \wedge \beta) > \Pi(\alpha \wedge \neg\beta)$. Moreover, hard rules of the form "all α 's are β 's" are represented by $\Pi(\alpha \wedge \neg\beta) = 0$. Hence, a default theory Υ can be

viewed as a family of constraints restricting a family $\prod(\Upsilon)$ of possibility distributions. Selecting a possibility distribution from $\prod(\Upsilon)$ using the minimum specificity principle defines the MSP-entailment which is equivalent to system P (Kraus, Lehmann, & Magidor 1990) augmented by rational monotony (Lehmann & Magidor 1992).

From a syntactical point of view, the MSP-entailment amounts to compute a possibilistic logic base. In fact, viewing a formula as expressing a rule with possible exceptions, the higher the certainty level of the formula, the more exceptional is the rule. Then, performing possibilistic inference on the obtained possibilistic base enables us to derive the MSP-conclusions of the given default theory. However, computing the possibilistic base is expensive from a computational point of view (a Δ_2^P problem). Moreover, performing inference on it is not tractable (a $\Delta_2^P[O(\log n)]$ -complete problem), which is problematic for applications, such as access control systems where queries need to be answered in polynomial time.

To address this problem, one can adhere to knowledge compilation. Knowledge compilation is a key direction of research in Artificial Intelligence (Cadoli & Donini 1997; Darwiche & Marquis 2002). It consists in preprocessing off-line the knowledge base in order to make the inference from it easier on-line.

In this paper, we propose to compile a default theory Υ with respect to a given target compilation language COMP . Let $\text{COMP}(\Upsilon)$ denotes the compiled theory. Then, we show that this compilation enables us to perform in polynomial time MSP-entailment. In addition, the proposed compilation is qualified to be flexible since it can take advantage of any classical compiler. Moreover, it provides an interesting recompilation cost when updating the default theory.

The rest of the paper is organized as follows. Section 2 gives backgrounds on possibility theory and possibilistic logic. In Section 3, we describe the possibilistic handling of default theories proposed in (Benferhat, Dubois, & Prade 1998). Knowledge compilation is recalled in Section 4. Our compilation approach is presented in Section 5. Finally, the last section concludes the paper.

Brief background on possibility theory and possibilistic logic

We consider a finite propositional language denoted by \mathcal{L} . Propositional variables are denoted by lower case Roman letters a, b, c, \dots and formulae by Greek letters $\alpha, \beta, \gamma, \dots$. The symbols \top and \perp denote tautology and contradiction respectively. A clause (resp. term) is a disjunction (resp. conjunction) of literals where a literal is a propositional variable or its negation.

The basic object of possibility theory (Zadeh 1978; Dubois & Prade 1988) is the *possibility distribution*, which is a mapping from the set of classical interpretations Ω to the interval $[0,1]$. More generally, the interval $[0,1]$ can be replaced by any bounded linearly ordered scale. A possibility distribution π represents the available knowledge about where the real world is. By convention, $\pi(\omega) = 1$ means that it is totally possible for ω to be the real world, $\pi(\omega) > 0$ means that ω is only somewhat possible, while $\pi(\omega) = 0$ means that ω is certainly not the real world. The inequality $\pi(\omega) > \pi(\omega')$ means that the situation ω is more possible than ω' . π induces two mappings grading respectively the possibility and the certainty of a formula α :

- The possibility measure $\Pi(\alpha) = \max\{\pi(\omega) : \omega \models \alpha\}$ which evaluates to what extent α is consistent with the available knowledge expressed by π .
- The certainty (or necessity) measure $N(\alpha) = 1 - \Pi(\neg\alpha)$ which evaluates to what extent α is entailed by the available knowledge expressed by π .

Another important notion in possibility theory is the *principle of minimal specificity*. A possibility distribution π is said to be less specific than another possibility distribution π' if and only if for each interpretation ω we have $\pi'(\omega) \leq \pi(\omega)$ and there exists at least one interpretation ω' such that $\pi'(\omega') < \pi(\omega')$.

Syntactically, possibility distributions are compactly represented by possibilistic logic knowledge bases (Dubois, Lang, & Prade 1994). A possibilistic logic formula is a pair made of a classical logic formula ψ and a weight $a \in (0, 1]$ expressing certainty. The weight a is interpreted as a lower bound of $N(\psi)$, i.e., the possibilistic logic expression (ψ, a) is understood as $N(\psi) \geq a$. In the following, $\Sigma_a = \{\psi_i : (\psi_i, a_i) \in \Sigma \text{ and } a_i \geq a\}$.

Definition 1 Let $\Sigma = \{(\psi_i, a_i) : i = 1, n\}$ be a possibilistic base. The inconsistency level of Σ , denoted by $\text{Inc}(\Sigma)$, is given by: $\text{Inc}(\Sigma) = \max\{a : \Sigma_a \models \perp\}$ (by convention, $\max\emptyset = 0$).

Three entailment relations (simple, weighted and possibilistic conditioning) can be defined from a possibilistic knowledge base as follows:

Definition 2 Let Σ be a possibilistic knowledge base. Let α and φ be two classical formulae. Then :

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

- φ is a consequence of conditioning Σ by α (resp. to a degree a_i) iff $\Sigma \cup \{(\alpha, 1)\} \models_{\pi} \varphi$ (resp. (φ, a_i)).

According to (Lang 2000), deciding the possibilistic inference is a $\Delta_2^p[O(\log n)]$ -complete problem: it needs at most $\log_2 n$ calls to a SAT solver. We refer the reader to (Papadimitriou 1994) for more details about computational complexity.

Possibilistic handling of default theories

By a piece of conditional assertion or a default rule we mean a generic rule of the form "generally, if α then β " having possibly some exceptions. These rules are denoted by " $\alpha \rightarrow \beta$ ". A default base is a set $\Delta = \{\alpha_i \rightarrow \beta_i : i = 1, n\}$ of default rules. The material implication is denoted by \Rightarrow and is used to encode strict rules (or hard rules) of the form "if δ_i is observed, then γ_i is always true". We denote by $W = \{\delta_i \Rightarrow \gamma_i : i = 1, m\}$ a set of such strict rules. Now, a default theory Υ is given by the pair (Δ, W) .

In (Benferhat, Dubois, & Prade 1998), it has been proposed to view each conditional assertion $\alpha \rightarrow \beta$ as a constraint expressing that the situation where α and β is true has a greater possibility than the one where α and $\neg\beta$ is true which is expressed in possibility theory by the strict inequality $\Pi(\alpha \wedge \beta) > \Pi(\alpha \wedge \neg\beta)$. Moreover, hard rules of the form "all α 's are β 's" are modelled by the condition $\Pi(\alpha \wedge \neg\beta) = 0$.

Hence, a default theory Υ can be viewed as a family of constraints $C(\Upsilon)$ restricting a family $\prod(\Upsilon)$ of possibility distributions. Elements of $\prod(\Upsilon)$ are said to be compatible with Υ .

A first way of defining a nonmonotonic consequence relation consists in considering all the possibility distributions of $\prod(\Upsilon)$, namely a conditional assertion $\alpha \rightarrow \beta$ is said to be a universal possibilistic consequence of Υ , denoted by $\Upsilon \models_{\forall\pi} \alpha \rightarrow \beta$, iff $\Pi(\alpha \wedge \beta) > \Pi(\alpha \wedge \neg\beta)$ for each possibility distribution π from $\prod(\Upsilon)$.

It has been shown in (Benferhat, Dubois, & Prade 1998) that in the case where $W = \emptyset$, the universal possibilistic inference relation $\models_{\forall\pi}$ is equivalent to System P (P as Preferential) which is a set of postulates encoded by a reflexivity axiom and five inference rules namely Left Logical Equivalence, Right Weakening, Or, Cautious Monotony and Cut (Kraus, Lehmann, & Magidor 1990).

However, the universal possibilistic consequence relation, even if it produces acceptable and safe conclusions, is very cautious. One way of coping with the cautiousness of $\models_{\forall\pi}$ is to pick only one possibility distribution among those in $\prod(\Upsilon)$. It is what it is done when the so-called rational monotony property (Lehmann & Magidor 1992) is added to system P. To this effect, the minimum specificity principle can be used (Benferhat, Dubois, & Prade 1998). The MSP-entailment (MSP as minimum specificity principle) is defined as follows:

Definition 3 Let π be a possibility distribution selected from $\prod(\Upsilon)$ using the minimum specificity principle. Then a conditional assertion $\alpha \rightarrow \beta$ is said to be a MSP-consequence of Υ , denoted by $\Upsilon \models_{MSP} \alpha \rightarrow \beta$, iff $\Pi(\alpha \wedge \beta) > \Pi(\alpha \wedge \neg\beta)$.

From a syntactic point of view, the MSP-entailment comes down to first compute a possibilistic base Σ_{Υ} as shown by Algorithm 1. This later extends the one developed in System Z (Pearl 1990) to the case where we both deal with hard and default rules. It basically consists to assign to each default rule a degree. Default rules with the lowest degree are the most general ones. They are such that assigning their antecedent to be true does not cause inconsistencies. Hence, we need:

- to determine which default rules are tolerated,
- to remove these tolerated default rules in order to stratify the remaining rules.

Once the possibilistic base has been generated, we apply a possibilistic inference. In fact, it has been proved that $\Upsilon \models_{MSP} \alpha \rightarrow \beta$ iff $\Sigma_{\Upsilon} \cup \{(\alpha, 1)\} \models_{\pi} \beta$, i.e. β is a conclusion of conditioning Σ_{Υ} by α .

Algorithm 1: Syntactic counterpart of MSP

Data: a default theory $\Upsilon = (\Delta, W)$

Result: a possibilistic base Σ_{Υ}

begin

```

  m ← 1;
  while Δ ≠ ∅ do
    A ← {¬αi ∨ βi : αi → βi ∈ Δ};
    Sm ← {αi → βi : αi → βi ∈ Δ and A ∪ W ∪ {αi} is consistent}
    if Sm = ∅ then
      ⊥ Stop the algorithm
    Δ ← Δ − Sm;
    m ← m + 1;
  ΣΥ ← Σ1 ∪ ... ∪ Σm where Σm = {(ϕ, 1) : ϕ ∈ W}
  and for j = 1, m − 1: Σj = {(¬αk ∨ βk, aj) : αk → βk ∈ Sj and aj = j/m}
  return ΣΥ;

```

end

Let us illustrate Algorithm 1 on the following example which expresses a simple security policy in a medical context:

Example 1 We consider the following set of rules:

- "all surgeons are doctors."
- "generally, surgeons can write surgical operations reports."
- "generally, doctors can not write surgical operations reports."

which can be symbolically written as $\Upsilon = (\Delta, W)$ where $\Delta = \{s \rightarrow w, d \rightarrow \neg w\}$, and $W = \{s \Rightarrow d\}$.

- At first step, $A = \{\neg s \vee w, \neg d \vee \neg w\}$. $A \cup \{d\} \cup W$ is consistent while $A \cup \{s\} \cup W$ is not so $S_1 \leftarrow \{d \rightarrow \neg w\}$.
- At the next step, $A = \{\neg s \vee w\}$. $A \cup \{s\} \cup W$ is consistent thus we get $S_2 = \{s \rightarrow w\}$.

So, $\Sigma_{\Upsilon} = \{(\neg d \vee \neg w, 1/3), (\neg s \vee w, 2/3), (\neg s \vee d, 1)\}$.

We are interested now in knowing if a given doctor who is a surgeon can write surgical operations reports or not. We

have $\Sigma_{\Upsilon} \cup \{(s \wedge d, 1)\} \models_{\pi} w$ since $\text{Inc}(\Sigma_{\Upsilon} \cup \{(s \wedge d, 1)\} \cup \{(\neg w, 1)\}) = 2/3 > \text{Inc}(\Sigma_{\Upsilon} \cup \{(s \wedge d, 1)\}) = 1/3$. Hence, $\Upsilon \models_{MSP} s \wedge d \rightarrow w$.

Knowledge Compilation

The key motivation of *Knowledge compilation* is that a knowledge base is not modified very often, and the same base is used to answer many queries (see (Cadoli & Donini 1997) for a survey). So, the idea in knowledge compilation is to split query answering into two phases:

- first, the knowledge base is preprocessed to obtain an appropriate data structure. Such a phase is called *off-line reasoning*.
- the second phase which is called *on-line reasoning*, consists in answering queries using the data structure generated during the first phase.

A target compilation language is a class of formulas which is tractable for clausal deduction at least. Recently, Darwiche and Marquis have considered in (Darwiche & Marquis 2002) a set of target compilation languages. These languages are special cases of the *NNF* (for Normal Negation Form) one obtained by imposing some properties. A NNF formula is a formula constructed with literals using only the conjunction and disjunction operators. As to the properties, one can list *decomposability*, *determinism*, *smoothness*, *decision*, *order*, etc.

The resulting target compilation languages are DNF, DNNF, d-DNNF, sd-DNNF, FBDD, OBDD, OBDD_<, MODS, PI and IP. Additionally, they are compared in terms of their spacial efficiency via the *succinctness* criteria and also in terms of the set of logical operations they support in polynomial time.

With the exception of PI, DNNF is the most succinct among all target compilation languages. In fact, it is known that PI is not more succinct than DNNF, but it is unknown whether DNNF is more succinct than PI.

A DNNF formula (for Decomposable NNF) is a NNF formula satisfying the decomposability property: for any conjunction $\bigwedge_i \alpha_i$ appearing in the formula, no variables are shared by the conjuncts α_i (Darwiche 2001).

We need now to define the operation of conditioning (which is different from the possibilistic conditioning) that will be useful in the following.

Definition 4 The conditioning of a propositional formula ψ on a consistent term γ , denoted by $\psi|\gamma$, is the formula obtained from ψ by substituting every literal in ψ that shares a variable with γ by \top if it is consistent with γ , by \perp otherwise.

Example 2 Let $\psi = (\neg a \wedge \neg b) \vee (b \wedge c)$.

- $\psi|(\neg a \wedge c) \equiv (\top \wedge \neg b) \vee (b \wedge \top) \equiv \top$.
- $\psi|(a \wedge b) \equiv (\perp \wedge \perp) \vee (\top \wedge c) \equiv c$.

Following (Darwiche & Marquis 2002), for any target compilation language COMP considered in the same paper, COMP satisfies conditioning. This means that given a formula ψ from COMP and any consistent term γ , we can construct in polynomial time a formula equivalent to $\psi|\gamma$ and

which belongs to COMP. This result will be very useful to ensure the efficiency of our compilation approach.

Compiling possibilistic default theories

In many applications, like modeling access control security policies, answering to queries (namely deriving plausible conclusions) needs to be performed in polynomial time. For instance, in ORBAC (for Organization Based Access Control) (AbouElKalam *et al.* 2003), W contains integrating constraints on role's hierarchies, separation of roles (a user can not activate at the same time the doctor's role and the role of director of hospital) ... Δ contains rules with exceptions on permissions's assignment to roles, user's assignment to roles (e.g., John can play role doctor). Rules of Δ are simple rules that can be easily put in a CNF form. Queries have simple forms that can be represented by terms. For instance, given the fact "it is night, patient's record P_1 is classified secret, John plays its role of doctor", we need to know if "John can read P_1 ".

The following is based on the assumptions that rules can be easily put in CNF form.

In order to derive MSP-conclusions from a default theory $\Upsilon = (\Delta, W)$ in polynomial time, a classical way consists first in computing the corresponding possibilistic knowledge base Σ_Υ using Algorithm 1 and then compiling the resulting possibilistic base according to a given possibilistic compilation method (see for instance (Benferhat, Yahi, & Drias 2007)) to ensure polynomial possibilistic inference.

However, Algorithm 1 is computationally expensive. Indeed, the decision version of the stratification problem represents a Δ_2^P problem. Hence, the classical way comes down to solve a Δ_2^P problem followed by a possibilistic base compilation process which is expensive too. Moreover, when the default theory changes (by adding or removing a formula), the whole process must be applied again and obviously the cost is very high.

In order to cope with these problems, we propose a direct and flexible compilation of a default theory which at once enables us to compute efficiently the associated possibilistic base and to perform possibilistic inference efficiently too. Consequently, such a compilation provides an efficient way to perform MSP-entailment in polynomial time. This is described in the following subsections.

Encoding the base

Our approach starts by encoding the default theory $\Upsilon = (\Delta, W)$ under the form of a classical propositional base as given by the following definition:

Definition 5 Given a default theory $\Upsilon = (\Delta, W)$ with $\Delta = \{\alpha_i \rightarrow \beta_i : i = 1, n\}$ and $W = \{\delta_i \Rightarrow \gamma_i : i = 1, m\}$. Let A_i 's ($i = 0, n$) be new propositional variables. The propositional encoding of Υ , denoted by K_Υ , is given by:

$$K_\Upsilon = \{\neg\alpha_i \vee \beta_i \vee A_i : \alpha_i \rightarrow \beta_i \in \Delta\} \cup \{\neg\delta_i \vee \gamma_i \vee A_0 : \delta_i \Rightarrow \gamma_i \in W\}.$$

In fact, our encoding amounts to replace each default rule $\alpha_i \rightarrow \beta_i$ by a classical propositional formula $\neg\alpha_i \vee \beta_i \vee A_i$ (for $i = 1, n$). As to the strict rules in W , we replace each $\delta_i \Rightarrow \gamma_i$ ($i = 1, m$) by the formula $\neg\delta_i \vee \gamma_i \vee A_0$.

Once we define the classical propositional base K_Υ , we compile it into a target compilation language COMP. This result corresponds to the compilation of Υ with respect to COMP given formally by:

Definition 6 Let $\Upsilon = (\Delta, W)$ be a default theory and COMP be a target compilation language. Let K_Υ be the associated propositional encoding according to Definition 5. The compilation of Υ with respect to COMP, denoted by $\text{COMP}(\Upsilon)$, is given by the compilation of the classical base K_Υ into the language COMP.

It is to note that the most of existing propositional compilers require the formula that we want to compile to be given under a conjunctive normal form (CNF), i.e, a conjunction of clauses. Thus, we first need to put (if it is not the case) $\neg\alpha_i$'s, β_i 's, $\neg\delta_i$'s and γ_i 's in a CNF form and then the CNF of K_Υ can be defined.

Example 3 We consider the default theory $\Upsilon = (\Delta, W)$ given by Example 1. Let A_0, A_1 and A_2 be three new propositional variables.

The propositional encoding of Υ is $K_\Upsilon = (\neg s \vee d \vee A_0) \wedge (\neg s \vee w \vee A_1) \wedge (\neg d \vee \neg w \vee A_2)$.

A compilation of Υ with respect to DNNF is the classical DNNF compilation of K_Υ , i.e, $\text{DNNF}(\Upsilon) = [s \wedge [(w \wedge ((A_0 \wedge \neg d) \vee (A_2 \wedge d))) \vee (A_1 \wedge \neg w \wedge (A_0 \vee d))]] \vee [\neg s \wedge (A_2 \vee \neg d \vee \neg w)]$. A DNNF compiler can be found in (Darwiche 2001).

Computing the associated possibilistic base

To determine the set of general rules, we first compute $K = \text{COMP}(\Upsilon) | (\bigwedge_{i \in I} \neg A_i)$ where initially $I = \{0, \dots, n\}$ which is equivalent to $W \cup \Delta$. In fact, the formula $(\neg\alpha_i \vee \beta_i \vee A_i) | \neg A_i \equiv \neg\alpha_i \vee \beta_i \vee \perp \equiv \neg\alpha_i \vee \beta_i$ while $(\neg\alpha_i \vee \beta_i \vee A_i) | A_i \equiv \neg\alpha_i \vee \beta_i \vee \top \equiv \top$ which comes down to remove the corresponding rule.

Now, $\alpha_i \rightarrow \beta_i \in \Delta$ is tolerated if $\{\alpha_i\} \cup \Delta \cup W$ is consistent. Hence it is enough to check whereas $K \not\models \neg\alpha_i$ holds. This test is performed in polynomial time since K belongs to the target compilation language COMP and since α_i 's are assumed to be in a conjunctive normal form.

So let I_m (where $m = 1$ at the beginning and incremented at each iteration) contains the indices of default rules from Δ that satisfy this condition.

In the next iteration, we consider only formulas that have not been assigned a degree, i.e, $(\neg\alpha_i \vee \beta_i)$'s where $i \in I$ and drop the other formulas. This amounts to compute $K \leftarrow \text{COMP}(\Upsilon) | (\bigwedge_{i \in I} \neg A_i \wedge \bigwedge_{i \in \bar{I}} A_i)$ where \bar{I} denotes the set $\{1, \dots, n\} - I$. At the end, $I_m \leftarrow \{0\}$ representing the set of rules without exceptions, i.e, W .

Algorithm 2 describes more formally this process.

Let $\Sigma'_\Upsilon = \Sigma_1 \cup \dots \cup \Sigma_m$ where $\Sigma_i = \{(\neg\alpha_k \vee \beta_k, a_i) : \alpha_k \rightarrow \beta_k \in \Delta \text{ and } k \in I_i \text{ and } a_i = i/m\}$ for $i < m$ and $\Sigma_m = \{(\neg\delta_k \vee \gamma_k, 1) : \delta_k \Rightarrow \gamma_k \in W\}$. Then the following proposition is valid:

Proposition 1 The possibilistic base Σ'_Υ corresponds exactly to the possibilistic base Σ_Υ given by Algorithm 1. In addition, Algorithm 2 runs in polynomial time.

Algorithm 2: Generating the associated possibilistic base
 Data: $\text{COMP}(\Upsilon)$ (A compilation of Υ w.r.t. COMP)
 Result: a sequence (I_1, \dots, I_m)

```

begin
   $m \leftarrow 1$ ;
   $I = \{i : 0 \leq i \leq n\}$ ;
   $K \leftarrow \text{COMP}(\Upsilon) | (\bigwedge_{i \in I} \neg A_i)$ ;
  while  $I \neq \{0\}$  do
     $I_m = \{i \in I - \{0\} : K \not\models \neg \alpha_i\}$ ;
    if  $I_m = \emptyset$  then
       $\perp$  Stop the algorithm;
     $I \leftarrow I - I_m$ ;
     $K \leftarrow \text{COMP}(\Upsilon) | (\bigwedge_{i \in I} \neg A_i \wedge \bigwedge_{i \in \bar{I}} A_i)$ ;
     $m \leftarrow m + 1$ ;
   $I_m \leftarrow \{0\}$ ;
  return  $(I_1, \dots, I_m)$ ;
end

```

Let us now illustrate Algorithm 2 on the following example:

Example 4 Let us consider Example 3 where $\text{DNNF}(\Upsilon) \equiv [s \wedge [(w \wedge ((A_0 \wedge \neg d) \vee (A_2 \wedge d))) \vee (A_1 \wedge \neg w \wedge (A_0 \vee d))]] \vee [\neg s \wedge (A_2 \vee \neg d \vee \neg w)]$.

- Initially, we have $m \leftarrow 1$, $I \leftarrow \{0, 1, 2\}$ and $K \leftarrow \text{DNNF}(\Upsilon) | (\neg A_0 \wedge \neg A_1 \wedge \neg A_2) \equiv \neg s \wedge (\neg d \vee \neg w)$.
- At first iteration, $K \models \neg s$ while $K \not\models \neg d$. So $I_1 \leftarrow \{2\}$ where 2 represents the indices in Δ of the default rule $d \rightarrow \neg w$. Moreover $K \leftarrow \text{DNNF}(\Upsilon) | (\neg A_0 \wedge \neg A_1 \wedge A_2) \equiv (s \wedge w \wedge d) \vee \neg s$.
- At second iteration, $K \not\models \neg s$ thus $I_2 \leftarrow \{1\}$. $I \leftarrow \{0\}$ which stops the loop.
- $I_3 \leftarrow \{0\}$.
- So, Algorithm 2 returns (I_0, I_1, I_2) from which Σ'_Υ is given by: $\Sigma'_\Upsilon = \Sigma_1 \cup \Sigma_2 \cup \Sigma_3$ where $\Sigma_1 = \{(\neg d \vee \neg w, 1/3)\}$, $\Sigma_2 = \{(\neg s \vee w, 2/3)\}$ and $\Sigma_3 = \{(\neg s \vee d, 1)\}$.

One can easily see that Σ'_Υ computed in polynomial time here corresponds exactly to Σ_Υ of Example 1.

Deriving MSP-conclusions

We show now how one can again take advantage of the compilation of Υ , $\text{COMP}(\Upsilon)$, for the sake of deriving in polynomial time MSP-consequences from Υ as described by Algorithm 3.

Proposition 2 Given a conditional assertion $\alpha \rightarrow \beta$, Algorithm 3 checks in polynomial time whether $\Upsilon \models_{\text{MSP}} \alpha \rightarrow \beta$ or not.

The idea is as follows. First, let $S_i = \{\neg \alpha_k \vee \beta_k : \alpha_k \rightarrow \beta_k \in \Delta \text{ and } k \in I_i\}$ for $i = 1, m-1$ and $S_m = W$. One can check that $\Upsilon \models_{\text{MSP}} \alpha \rightarrow \beta$ iff $\Sigma'_\Upsilon \cup \{(\alpha, 1)\} \models_\pi$

Algorithm 3: Computing MSP-conclusions
 Data: $\text{COMP}(\Upsilon)$ (A compilation of Υ w.r.t. COMP),
 (I_1, \dots, I_m) given by Algorithm 2,
 a conditional assertion $\alpha \rightarrow \beta$

```

begin
   $i \leftarrow m$ ;
   $\text{IsConclusion} \leftarrow \text{false}$ ;
   $\text{IsSat} \leftarrow \text{true}$ ;
  while  $(i \geq 1 \text{ and } \text{IsSat} \text{ and } \neg \text{IsConclusion})$  do
     $\gamma \leftarrow (\bigwedge_{k \in X_i} \neg A_k \wedge \bigwedge_{k \in \bar{X}_i} A_k)$  where  $X_i = \bigcup_{j=i}^m I_j$ ;
     $K \leftarrow \text{COMP}(\Upsilon) | \gamma$ ;
    if  $K \models \neg \alpha$  then
       $\text{IsSat} \leftarrow \text{false}$ 
    else
      if  $K \models \neg \alpha \vee \beta$  then
         $\text{IsConclusion} \leftarrow \text{true}$ ;
      else
         $i \leftarrow i - 1$ ;
  if  $\text{IsConclusion} = \text{true}$  then
     $\Upsilon \models_{\text{MSP}} \alpha \rightarrow \beta$ 
  else
     $\Upsilon \not\models_{\text{MSP}} \alpha \rightarrow \beta$ 
end

```

β iff $\exists i \geq 1$ such that $\bigcup_{j=i}^m S_j \cup \{(\alpha, 1)\}$ is consistent and $\bigcup_{j=i}^m S_j \cup \{(\alpha, 1)\} \models \beta$. On the other hand $\bigcup_{j=i}^m S_j$ can be obtained in polynomial time from $\text{COMP}(\Upsilon)$ by conditioning it on the term $(\bigwedge_{k \in X_i} \neg A_k \wedge \bigwedge_{k \in \bar{X}_i} A_k)$ where $X_i = \bigcup_{j=i}^m I_j$.

Example 5 Let us consider Example 4 and let us apply Algorithm 3 to check again if a given doctor who is a surgeon is permitted or not to write surgical operations reports.

- At first iteration, $i \leftarrow 3$ so $K \leftarrow \text{COMP}(\Upsilon) | (\neg A_0 \wedge A_1 \wedge A_2) \equiv [s \wedge ((w \wedge d) \vee (\neg w \wedge d))] \vee \neg s$. We have $K \not\models \neg s \vee \neg d$ and $K \not\models (\neg s \vee \neg d) \vee w$.
- At second iteration, $K \leftarrow \text{COMP}(\Upsilon) | (\neg A_0 \wedge \neg A_1 \wedge A_2) \equiv (s \wedge w \wedge d) \vee \neg s$. We have $K \not\models \neg s \vee \neg d$ and $K \models (\neg s \vee \neg d) \vee w$. So we deduce that $\Upsilon \models_{\text{MSP}} s \wedge d \rightarrow w$ which means that the given surgeon is permitted to write surgical operations reports.

On the flexibility of our approach

Our approach is parameterized by any target compilation language since it relies only on CNF deduction (hence clausal deduction) and conditioning operations. Now, we are interested in its behavior when an already compiled default theory changes either by removing an existing default

rule or by adding a new one. Stated differently, what is the corresponding recompilation cost ?

- Let us first consider the case where we want to delete a default rule $\alpha_i \rightarrow \beta_i$. One can easily see that this comes down to condition $\text{COMP}(\Upsilon)$ on A_i and then apply Algorithm 2. These two steps can be achieved in polynomial time and $\text{COMP}(\Upsilon)$ still belongs to the language COMP . Consequently, we do not need any recompilation.
- As to the case of adding a new rule $\alpha_s \rightarrow \beta_s$, let us first recall the following point. It is well known (Darwiche & Marquis 2002) that if we take two formulas ψ and χ from a target compilation language COMP , a polytime algorithm for computing a formula equivalent to their conjunction $\psi \wedge \chi$ and which belongs to COMP exists only for $\text{COMP} \in L$ where $L = \{DNF, OBDD_<, IP, MODS\}$. So, in the case where $\text{COMP} \in L$, it is sufficient to compile the formula $\neg\alpha_s \vee \beta_s \vee A_s$ into COMP and then making its conjunction with $\text{COMP}(\Upsilon)$. Obviously, the recompilation cost here is clearly better than recompiling the whole new default theory.

Conclusion

In this paper, we have proposed a compilation approach of default theories. This approach presents several advantages from a computational point of view.

Indeed, the compiled default theory allows us to perform MSP-entailment (which is equivalent to system P augmented by rational monotony) in polynomial time.

On the other hand, this approach is parameterized by any target compilation language which makes it flexible. In addition, updating a compiled default theory does not imply any recompilation in the case of removing an existing default rule. As for the case of inserting a new rule, the recompilation cost depends on the target compilation language in question. In fact, this recompilation cost is too negligible with certain target compilation languages.

A future work is to extend this approach in order to handle the lexicographic closure (Lehmann 1995) which has been shown to be very satisfactory from a psychological view point.

References

- AbouElKalam, A.; Benferhat, S.; Miège, A.; ElBaida, R.; Cuppens, F.; Saurel, C.; Balbiani, P.; Deswarte, Y.; and Trouessin, G. 2003. Organization based access control. In *POLICY*, 120–128.
- Benferhat, S.; Dubois, D.; and Prade, H. 1998. Practical handling of exception-tainted rules and independence information in possibilistic logic. *Journal of Applied Intelligence* 9(2):101–127.
- Benferhat, S.; Yahi, S.; and Drias, H. 2007. On the compilation of stratified belief bases under linear and possibilistic logic policies. In *Proceedings of IJCAI' 07*, 2428–2430.
- Cadoli, M., and Donini, F. M. 1997. A survey on knowledge compilation. *AI Communications* 10:137–150.
- Darwiche, A., and Marquis, P. 2002. A knowledge compilation map. *Journal of Artificial Intelligence Research* 17:229–264.
- Darwiche, A. 2001. Decomposable negation normal form. *Journal of the ACM* 48(4):608–647.
- Dubois, D., and Prade, H. 1988. *Possibility Theory: An Approach to Computerized Processing of Uncertainty*. New York: Plenum Press.
- Dubois, D.; Lang, J.; and Prade, H. 1994. Possibilistic logic. *Handbook of Logic in Artificial Intelligence and Logic Programming* 3:439–513.
- Kraus, S.; Lehmann, D. J.; and Magidor, M. 1990. Non-monotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence* 44(1-2):167–207.
- Lang, J. 2000. Possibilistic logic: Complexity and algorithms. *Handbook of Defeasible Reasoning and Uncertainty Management Systems* 5:179–220.
- Lehmann, D. J., and Magidor, M. 1992. What does a conditional knowledge base entail? *Artificial Intelligence* 55(1):1–60.
- Lehmann, D. J. 1995. Another perspective on default reasoning. *Annals of Mathematics and Artificial Intelligence* 15(1):61–82.
- Morgenstern, L. 1997. Inheritance comes of age: applying nonmonotonic techniques to problems in industry. In *Proceedings of IJCAI' 97*, 1613–1613.
- Papadimitriou, C. H. 1994. *Computational Complexity*. Addison Wesley Publishing Company.
- Pearl, J. 1990. System z: A natural ordering of defaults with tractable applications to default reasoning. In *Proceedings of TARK'90*, 121–135.
- Zadeh, L. A. 1978. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28.