JÉRÔME LANG

POSSIBILISTIC LOGIC: COMPLEXITY AND ALGORITHMS

1 INTRODUCTION

Possibilistic logic is a logic of uncertainty tailored for reasoning under incomplete and partially inconsistent knowledge. At the syntactical level it handles formulae of propositional or first-order classical logic, to which are attached lower bounds of so-called degrees of necessity and possibility. The degree of necessity (resp. possibility) of a formula expresses to what extent the formula is entailed by (resp. compatible with) the available evidence. At the mathematical level, degrees of possibility and necessity are closely related to fuzzy sets [Zadeh, 1965; Zadeh, 1978] and possibilistic logic is especially adapted to automated reasoning when the available information is pervaded with vagueness. A vague piece of evidence can be viewed as defining an implicit ordering on the possible worlds it refers to, this ordering being encoded by means of fuzzy set membership functions. Hence, possibilistic logic is a tool for reasoning under uncertainty based on the idea of (complete) ordering rather than counting, unlike probabilistic logic. For a complete exposition of possibilistic logic see [Dubois *et al.*, 1994]; a more general introduction to possibility theory and is in [Dubois and Prade, 1988].

Going now deeper into the formal details, possibilistic logic in its simplest version ("standard possibilistic logic") considers *certainty-valued statements* of the form $N(\varphi) \geq \alpha$, where φ is a well-formed formula of a classical propositional or first-order language $\mathcal{L}$, $\alpha \in (0,1]$ and $N(\varphi)$ the necessity degree of φ. These statements will be expressed by the syntactical object $(\varphi\ \alpha)$ and will be called *possibilistic formulae*. Thus, $(\varphi\ \alpha)$ acts as a constraint on the set of necessity measures: a necessity measure N on the language $\mathcal{L}$ satisfies (φ, α) iff $N(\varphi) \geq \alpha$. More generally, a set of possibilistic formulae $(\varphi_i\ \alpha_i)$ logically entails a possibilistic formula $(\psi\ \beta)$ iff any necessity measure satisfying $\forall i, N(\varphi_i) \geq \alpha_i$ satisfies also $N(\psi) \geq \beta$. The basic *deduction problem* in possibilistic logic consists then in finding the greatest β such that $(\psi\ \beta)$ is logically entailed by the available knowledge composed by a set of possibilistic formulae.

It is worth noticing that possibilistic logic can be viewed as a "two-level logic". At the lower level, well-formed formulae of $\mathcal{L}$ are not only formed on a classical language, but they really *are* formulae from classical (propositional or first-order) logic from a semantical point of view. At the upper level, the necessity (and possibility) degrees weighing the formulae respect completely the structure of classical logic. Clearly, other similar logics can be obtained by replacing the possibility and necessity measures by other classes of functions (for instance probabilities). The variety of logics thus obtained by letting these classes of functions vary will be called *weighted logics*. The only requirement we ask for a weighted logic is that the mappings g of the associated class map the language $\mathcal{L}$ into an ordered structured, and that they respect the structure of classical logic, *i.e.*, for any two logically equivalent formulae φ and φ' then $g(\varphi) = g(\varphi')$. Now, among the set of all weighted logics, we are especially interested in a distinguished subset (to which possibilistic logic belongs to), namely, *weighted logics of uncertainty*. Weighted logics of uncertainty not only respect classical equivalence but also require the *Sugeno property*, or monotonicity with respect to classical entailment:

$$\text{if } \varphi \text{ logically entails } \psi \text{ then } g(\varphi) \leq g(\psi)$$

This condition ensures that g can be interpreted as a degree of uncertainty.

At this point we should insist on the following point: *possibilistic logic is not a multiple-valued logic*. Generally speaking, multiple-valued logics are non-classical logics whose semantics are defined in terms of truth values belonging to a totally ordered set. Multiple-valued logics being non-classical logics mean that the set of classical tautologies is not identical to the set of tautologies of a multiple-valued logic – this is the case for instance for $\varphi \vee \neg\varphi$, or $\neg(\varphi \wedge \neg\varphi)$. Conversely, possibilistic logic is rather a meta-logic built on classical logic than a non-classical logic strictly speaking – since it respects completely the structure of classical logic. More generally, multiple-valued logics are generally dedicated to the representation of partial truth (where a degree of truth is viewed as the measure of compatibility of a vague statement with reality), while logics of uncertainty (and among them possibilistic logic) are dedicated to the representation of states of partial ignorance. For a discussion about degrees of truth and degrees of uncertainty see [Dubois and Prade, 1988; Dubois *et al.*, 1991a].

This chapter contains a comprehensive study of algorithmic and complexity issues related to possibilistic logic. In Section 2 we recall the basics of possibility theory. Section 3 discusses algorithmic issues for possibilistic deduction, which mainly consists in a possibilistic extension of refutation by resolution. Section 4 discusses algorithmic issues for possibilistic model finding, based on an extension of the procedure of Davis and Putnam. While Sections 3 and 4 focus only on the simplest fragment of possibilistic logic ("necessity-valued logic"), handling only certainty-valued statements, Section 5 discusses proof methods for an extended fragment of possibilistic logic, which handles both certainty-valued statements and possibility-valued statements. Section 6 concludes and points to work on related subjects.

2 STANDARD POSSIBILISTIC LOGIC: FORMAL BACKGROUND

In full possibilistic logic (discussed in Section 5), uncertain knowledge is expressed in terms of certainty- and possibility-qualified statements; "full" possibilistic logic handles syntactic objects expressing inequalities resulting from these statements. These objects, called possibilistic formulae, are the basic objects of full possibilistic logic. The so-called "standard", or necessity-valued, fragment of possibilistic logic (SPL) handles only possibilistic formulae corresponding to certainty-qualified statements. Although this fragment is poorer than full possibilistic logic, there are two good reasons for studying it separately before considering possibilistic logic in its full version: first, algorithmic issues are simpler to present in the necessity-valued fragment, and then their extension to the full case can be easily understood; second, from a knowledge representation point of view, the necessity-valued fragment SPL is significant, since it is sufficient for modelling a preference order upon formulae, and, as such, it entertains close links with the nonmonotonic approach based on preferential models and belief revision theory (see [Dubois and Prade, 1991b; Dubois and Prade, 1992]).

This Section borrows much material from [Dubois *et al.*, 1994] and [Lang, 1991a]. Proofs of the results that are not directly related to algorithmical and complexity issues are omitted – most of them can be found in [Dubois *et al.*, 1994].

2.1 Language

In the rest of the chapter, $\mathcal{L}$ is a propositional or first-order logical language (restricted to closed well-formed formulae), equipped with the semantics of classical logic. $\models$ denotes classical entailment. Well-formed formulae will be denoted by φ, ψ, etc. Classical worlds will be denoted by ω, ω', etc, and the set of all possible worlds by Ω. $\top$ and $\bot$ denote respectively tautology and contradiction.

A *necessity-valued formula* (also called SPL formula) is a pair $(\varphi\ \alpha)$, where φ is a classical propositional or first-order formula of $\mathcal{L}$ and $\alpha \in (0,1]$ is a positive number. $(\varphi\ \alpha)$ expresses that φ is certain at least to the degree α, *i.e.*, $N(\varphi) \geq \alpha$, where N is a necessity measure modelling the state of knowledge of the agent. α is called the valuation of the formula and is denoted $val(\varphi)$.

A *necessity-valued knowledge base* (also called SPL knowledge base) $\mathcal{F}$ is then defined as a finite set (i.e. a conjunction) of necessity-valued formulae. SPL denotes the language consisting of necessity-valued formulae. $\mathcal{F}^*$ denotes the set of classical formulae obtained from $\mathcal{F}$ by ignoring the weights: if $\mathcal{F} = \{(\varphi_i, \alpha_i), i = 1 \ldots n\}$, then $\mathcal{F}^* = \{\varphi_i, i = 1 \ldots n\}$. It is called the *classical projection* of $\mathcal{F}$.

A SPL knowledge base may also be seen as a collection of nested sets of classical formulae: α being any valuation of $(0,1]$, we define the *α-cut* $\mathcal{F}_\alpha$ and the *strict α-cut* $\mathcal{F}_{\bar{\alpha}}$ by

$$\mathcal{F}_\alpha = \{(\varphi\ \beta) \in \mathcal{F} | \beta \geq \alpha\}$$

$$\mathcal{F}_{\bar{\alpha}} = \{(\varphi\ \beta) \in \mathcal{F} | \beta > \alpha\}$$

Their classical projections are thus

$$\mathcal{F}^*_\alpha = \{\varphi | (\varphi\ \beta) \in \mathcal{F}, \beta \geq \alpha\}$$

$$\mathcal{F}^*_{\bar{\alpha}} = \{\varphi | (\varphi\ \beta) \in \mathcal{F}, \beta > \alpha\}$$

Thus, a SPL knowledge base $\mathcal{F}$ can be viewed as a layered knowledge base, where the higher levels (α close to 1) correspond to the most certain pieces of knowledge. Reasoning from such a knowledge base will aim at deriving conclusions by means of the most certain parts of $\mathcal{F}$.

Interestingly, the two-level view of possibilistic logic suggests that valuations appear as *labels* associated to formulae; thus, possibilistic logic can be cast in Gabbay's Labelled Deductive Systems framework [Gabbay, 1991], where the set of labels is the totally ordered set $[0, 1]$ and the operations defined on it follow directly from the axioms of possibility theory. See [Dubois *et al.*, 1994] for connections with other general frameworks.

2.2 Semantics and partial inconsistency

In standard possibilistic logic, satisfaction and logical consequence are defined by means of possibility distributions on the set of classical worlds Ω. A *possibility distribution* π is a function from Ω to $[0, 1]$. $\pi(\omega)$ reflects how possibly ω is the real world. When $\pi(\omega) = 1$ (resp. $\pi(\omega) = 0$), it is completely possible (resp. completely impossible) that ω is the real world. A possibility distribution π is *normalized* iff $\exists\omega$ such that $\pi(\omega) = 1$. In this paper we do not assume that possibility distributions are necessarily normalized.

Satisfaction of a knowledge base by a possibility distribution

The *possibility measure* Π induced by π is a function from $\mathcal{L}$ to $[0, 1]$ defined by

$$\Pi(\varphi) = sup\{\pi(\omega), \omega \models \varphi)\}$$

The dual *necessity measure* N induced by π is defined by

$$N(\varphi) = 1 - \Pi(\neg\varphi) = inf\{1 - \pi(\omega), \omega \models \neg\varphi)\}$$

Giving up the normalization condition $sup\{\pi(\omega), \omega \in \Omega\} = 1$ slightly modifies the behaviour of necessity measures with respect to the usual possibility theory: if $\alpha_\pi = 1 - sup\{\pi(\omega), \omega \in \Omega\}$, then we have

$$\forall\varphi, min(N(\varphi), N(\neg\varphi)) = \alpha_\pi > 0$$

which leads to $N(\bot) = N(\varphi \wedge \neg\varphi) = min(N(\varphi), N(\neg\varphi)) = \alpha_\pi$ instead of $N(\bot) = 0$. However, the following properties still hold:

$$N(\top) = 1;$$

$$N(\varphi \wedge \psi) = min(N(\varphi), N(\psi));$$

$$N(\varphi \vee \psi) \geq max(N(\varphi), N(\psi));$$

$$\mathrm{i}f\ \varphi \models \psi\ then\ N(\psi) \geq N(\varphi)$$

A possibility distribution π is said to *satisfy* the SPL formula $(\varphi\ \alpha)$ iff $N(\varphi) \geq \alpha$, where N is the necessity measure induced by π. We shall then use the notation $\pi \models (\varphi\ \alpha)$. A possibility distribution ψ satisfies a SPL knowledge base $\mathcal{F} = \{(\varphi_i\ \alpha_i) | i = 1...n\}$ iff $\forall i, \pi \models (\varphi_i\ \alpha_i)$. This is denoted by $\pi \models \mathcal{F}$.

Logical consequences

A SPL formula $(\varphi\ \alpha)$ is a *logical consequence* of the SPL knowledge base $\mathcal{F}$ iff for any π satisfying $\mathcal{F}$, then π also satisfies $(\varphi\ \alpha)$. This deduction problem will then be stated in the following manner: let $\mathcal{F}$ be a SPL knowledge base and φ a classical formula that we would like to deduce from $\mathcal{F}$ to some degree; we have to compute the highest valuation α (i.e. the best lower bound of a necessity degree) such that $(\varphi\ \alpha)$ is a logical consequence of $\mathcal{F}$, i.e. to compute

$$Val(\varphi, \mathcal{F}) = sup\{\alpha \in (0, 1], \mathcal{F} \models (\varphi\ \alpha)\}$$

Principle of minimum specificity

A fundamental result about deduction from possibilistic knowledge bases in SPL is that there always exists a least specific, i.e. highest, possibility distribution satisfying a possibilistic knowledge base $\mathcal{F}$. Namely, if $\mathcal{F} = \{(\varphi_i\ \alpha_i), i = 1...n\}$, then the least specific possibility distribution $\pi_{\mathcal{F}}$ satisfying $\mathcal{F}$ is defined by

$$\pi_{\mathcal{F}}(\omega) = \begin{cases} \min\{1 - \alpha_i \mid \omega \models \neg\varphi_i, i = 1...n\} \\ 1 \text{ if } \omega \models \varphi_1 \wedge ... \wedge \varphi_n \end{cases}$$

PROPOSITION 1. *[Dubois* et al.*, 1994] For any possibility distribution π, π satisfies $\mathcal{F}$ if and only if $\pi \leq \pi_{\mathcal{F}}$, i.e. $\forall \omega, \pi(\omega) \leq \pi_{\mathcal{F}}(\omega)$.*

As a corollary,

PROPOSITION 2. *[Dubois* et al.*, 1994]*

$$\mathcal{F} \models (\varphi\ \alpha) \textit{ iff } \pi_{\mathcal{F}} \models (\varphi\ \alpha)$$

In other terms, $Val(\varphi, \mathcal{F}) = N_{\mathcal{F}}(\varphi)$, where $N_{\mathcal{F}}$ is the necessity measure induced by $\pi_{\mathcal{F}}$.

Partial inconsistency

One of the nice features of standard possibilistic logic is that it enables a gradation of inconsistency and a nontrivial notion of deduction from a partially inconsistent knowledge base. A possibilistic knowledge base $\mathcal{F}$ whose associated possibility distribution $\pi_{\mathcal{F}}$ is such that $0 < sup\ \pi_{\mathcal{F}} < 1$ is said to be *partially inconsistent*. Measuring the consistency of $\mathcal{F}$ consists, then, of evaluating to what degree there is at least one completely possible interpretation for $\mathcal{F}$, i.e. to what degree the set of possibility distributions satisfying $\mathcal{F}$ contains normalized possibility distributions; the quantity

$$Cons(\mathcal{F}) = sup_{\pi \models \mathcal{F}} sup_{\omega \in \Omega} \pi(\omega) = sup_{\omega \in \Omega} \pi_{\mathcal{F}}(\omega)$$

will be called *consistency degree* of $\mathcal{F}$. Its complement to 1,

$$Incons(\mathcal{F}) = 1 - sup_{\omega \in \Omega} \pi_{\mathcal{F}}(\omega)$$

is called the *inconsistency degree* of $\mathcal{F}$.

Example: let $\mathcal{F} = \{(p\ 0.8), (q\ 0.3), (p \rightarrow r\ 0.6), (q \rightarrow \neg r\ 0.9)\}$. We get

$$\begin{array}{llll} \pi_{\mathcal{F}}(pqr) = 0.1; & \pi_{\mathcal{F}}(pq\bar{r}) = 0.4; & \pi_{\mathcal{F}}(p\bar{q}r) = 0.7; & \pi_{\mathcal{F}}(p\bar{q}\bar{r}) = 0.4; \\ \pi_{\mathcal{F}}(\bar{p}qr) = 0.1; & \pi_{\mathcal{F}}(\bar{p}q\bar{r}) = 0.2; & \pi_{\mathcal{F}}(\bar{p}\bar{q}r) = 0.2; & \pi_{\mathcal{F}}(\bar{p}\bar{q}\bar{r}) = 0.2. \end{array}$$

Hence, $sup_{\omega \in \Omega} \pi_{\mathcal{F}}(\omega) = 0.7$ and thus $Incons(\mathcal{F}) = 0.3$.

Inconsistency degrees enable the gradation of inconsistency. The two extreme cases are $Incons(\mathcal{F}) = 0$ (complete consistency) and $Incons(\mathcal{F}) = 1$ (complete inconsistency). When $0 < Incons(\mathcal{F}) < 1$, $\mathcal{F}$ is *partially inconsistent*. A partially inconsistent knowledge base entails contradictions with a positive necessity degree, i.e., $\mathcal{F} \models (\perp\ \alpha)$ for some $\alpha > 0$. Namely,

PROPOSITION 3. *[Dubois* et al.*, 1994]*

$$Incons(\mathcal{F}) = inf\{N(\perp) | \pi \models \mathcal{F}\} = N_{\mathcal{F}}(\perp) = sup\{\alpha | \mathcal{F} \models (\perp\ \alpha)\}$$

This equality justifies the terminology "inconsistency degree".

Furthermore, it can be proved that $Incons(\mathcal{F})$ is the valuation of the least certain formula in the strongest contradiction in $\mathcal{F}$. Namely,

PROPOSITION 4. *[Dubois* et al.*, 1994]*

$$Incons(\mathcal{F}) = \max_{\mathcal{F}' \subseteq \mathcal{F}, \mathcal{F}' inconsistent} \min\{\alpha | (\phi\ \alpha) \in \mathcal{F}'\}$$

Deduction under partial inconsistency

Let $\mathcal{F}$ be a partially inconsistent SPL knowledge base (such that $0 < Incons(\mathcal{F}) < 1$). Since any formula ϕ we have $N(\phi) \geq N(\perp)$, any formula ϕ is deducible from $\mathcal{F}$ with a valuation greater or equal to $Incons(\mathcal{F})$. It means that any deduction such that $\mathcal{F} \models (\phi\ \alpha)$ with $\alpha \leq Incons(\mathcal{F})$ may be due only to partial inconsistency of $\mathcal{F}$ and perhaps has nothing to do with ϕ. These deductions are called *trivial deductions*; on the contrary, deductions of necessity-valued formulae $\mathcal{F} \models (\phi\ \alpha)$ with $\alpha > Incons(\mathcal{F})$ are not caused by the partial inconsistency; they are called *nontrivial deductions*:

$$\mathcal{F} \models\!\approx (\phi\ \alpha) \text{ iff } \mathcal{F} \models (\phi\ \alpha) \text{ and } \alpha > Incons(\mathcal{F})$$

While the operator $\models$ is monotonic, $\models\!\!\approx$ is not.

Thus, $Incons(\mathcal{F})$ acts as a *threshold* inhibiting all formulae of $\mathcal{F}$ with a valuation equal to or below it. The following result shows its role as a threshold for the deduction problem more deeply:

PROPOSITION 5. *[Dubois* et al.*, 1994] Let* $inc = Incons(\mathcal{F})$ *and* $\alpha > 0$. *Then*

1. $\mathcal{F}_{\overline{inc}}$ *is consistent*[1]*;*
2. $\mathcal{F} \models\!\!\approx (\phi\ \alpha)$ *iff* $\mathcal{F}_{\overline{inc}} \models (\phi\ \alpha)$.

This result shows that only the consistent part of $\mathcal{F}$ consisting of formulae with a weight strictly greater than the inconsistency degree is significant for the deduction process. The next result establishes a link between inconsistency degrees and inconsistency in classical logic and is thus central to the deduction problem.

PROPOSITION 6. *[Dubois* et al.*, 1994]*

$$Incons(\mathcal{F}) = sup\{\alpha \mid \mathcal{F}^*_\alpha \textit{ is inconsistent}\} = inf\{\alpha \mid \mathcal{F}^*_{\overline{\alpha}} \textit{ is consistent}\}$$

The following result generalizes the deduction and refutation theorems from classical to possibilistic logic:

PROPOSITION 7. *[Dubois* et al.*, 1994]*

$$\mathcal{F} \cup \{(\phi\ 1)\} \models (\psi\ \alpha) \textit{ iff } \mathcal{F} \models (\phi \to \psi\ \alpha)$$

PROPOSITION 8 (refutation). *[Dubois* et al.*, 1994]*

$$\mathcal{F} \models (\phi\ \alpha) \textit{ iff } \mathcal{F} \cup \{(\neg\phi\ 1)\} \models (\bot\ \alpha)$$

or, equivalently,

$$Val(\phi, \mathcal{F}) = Incons(\mathcal{F} \cup \{(\neg\phi\ 1)\})$$

This result shows that any deduction problem in possibilistic logic can be viewed as computing an inconsistency degree. Lastly, we give the following result, stating that in order to deduce $(\phi\ \alpha)$, only the formulae with a weight greater than or equal to α are useful:

PROPOSITION 9. *[Dubois* et al.*, 1994]*

$$\mathcal{F} \models (\phi\ \alpha) \textit{ iff } \mathcal{F}_\alpha \models (\phi\ \alpha)$$

Preferred models

As seen before, knowing the possibility distribution $\pi_\mathcal{F}$ is sufficient for any deduction problem in SPL (including the computation of the inconsistency degree). It

[1] Recall that $\mathcal{F}_{\overline{inc}}$ is the strict α-cut of $\mathcal{F}$ (defined in Section 2.1) for $\alpha = inc$.

is important to notice that not only $\pi_{\mathcal{F}}$ is the least specific distribution satisfying $\mathcal{F}$, but it also *minimizes inconsistency* among all possibility distributions satisfying $\mathcal{F}$, i.e. $N_{\mathcal{F}}(\bot) = Incons(\mathcal{F}) = inf\{N(\bot)|\pi \models \mathcal{F}\}$. Now, $\pi_{\mathcal{F}}$ defines a *fuzzy subset* of Ω, which can be seen as the fuzzy set of (classical) models of $\mathcal{F}$, its membership function being $\pi_{\mathcal{F}}(\omega)$. The quantity $\pi_{\mathcal{F}}(\omega)$ represents the *compatibility* degree of ω with $\mathcal{F}$, measuring to what degree ω is a model of $\mathcal{F}$. From a decision-theoretic perspective, possibilistic formulae can be viewed as prioritized constraints (the valuation being the priority) and $\pi_{\mathcal{F}}(\omega)$ is then the degree to which ω satisfies the set of fuzzy constraints expressed by $\mathcal{F}$. See [Lang, 1991b] for this interpretation of posssibilistic logic in terms of prioritized constraints, and also [Dubois *et al.*, 1996] for a similar interpretation but with a CSP-based formalism.

Thus, $\pi_{\mathcal{F}}$ defines a preordering relation on Ω, namely, $\omega \geq_{\mathcal{F}} \omega'$ iff $\pi_{\mathcal{F}}(\omega) \geq \pi_{\mathcal{F}}(\omega')$. The worlds maximizing $\pi_{\mathcal{F}}(\omega)$ are called the *best models* of $\pi_{\mathcal{F}}$. It can be proved that the set of best models of $\mathcal{F}$ is never empty, i.e., the least upper bound in the computation if $Incons(\mathcal{F})$ is reached. If $\mathcal{F}$ represents prioritized constraints, then the best models are those which minimize the level of the most important among the violated constraints, and thus represent the best decisions, i.e. the most compatible with the constraints in $\mathcal{F}$.

3 ALGORITHMS FOR DEDUCTION IN STANDARD POSSIBILISTIC LOGIC

In this section we investigate complexity and algorithmic issues related to the deduction problem in SPL. We consider in succession several versions of the deduction problem, as already evoked in Section 2. For complexity issues we restict the problem to *propositional* necessity-valued logic.

3.1 Problem formulations and complexity

There are two different formulations of the "standard" deduction problem in SPL given the possibilistic knowledge base $\mathcal{F}$ and a formula ψ, the *optimization formulation* of the deduction problem consists in computing the best possible lower bound of the necessity degree of ψ given $\mathcal{F}$, denoted by $Val(\phi, \mathcal{F})$; given $\mathcal{F}$, ϕ and $\alpha \in (0, 1]$, the *decision formulation* of the deduction problem consists in deciding whether $(\phi\ \alpha)$ is a logical consequence of $\mathcal{F}$.

DEFINITION 10 (standard deduction problem in SPL, optimization form).

$$\text{(SPL-OPT) compute } Val(\phi, \mathcal{F}) = sup\ \{\alpha \mid \mathcal{F} \models (\phi\ \alpha)\}$$

DEFINITION 11 (standard deduction problem, decision form).

$$\text{(SPL-DED): decide whether } \mathcal{F} \models (\phi\ \alpha)$$

Due to Proposition 8, SPL-OPT can be reduced to computing an inconsistency degree; the converse is trivial since $Incons(\mathcal{F}) = Val(\perp, \mathcal{F})$. Interestingly, the decision form of the deduction problem is not harder than deduction in classical propositional logic.

PROPOSITION 12. SPL-DED *is* coNP*-complete.*

Proof. Follows directly from Proposition 9. ■

Let us turn now to the complexity of SPL-OPT, which is a more interesting problem since it has a more practical impact.

PROPOSITION 13. *Let $\mathcal{F}$ be a* propositional *SPL knowledge base and ϕ a propositional formula. Then, computing $Val(\phi, \mathcal{F})$ is* NP*-hard and requires $\lceil \log_2 n \rceil$ satisfiability checks, where n is the number of different valuations involved in $\mathcal{F}$.*

Proof. Following Proposition 6 we have $Incons(\mathcal{F}) = sup\{\alpha \mid \mathcal{F}^*_\alpha$ is inconsistent$\}$; this result enables us to compute $Val(\phi, \mathcal{F})$ by dichotomy, using any prover for the propositional satisfiability problem SAT. Let $\alpha_0 = 0$ and let $\alpha_1, ..., \alpha_n$ be the distinct valuations appearing in $\mathcal{F}$, ranked increasingly : $0 < \alpha_1, ..., \alpha_n \leq 1$.

```
begin
    l := 0 ;
    u := n ;
    while l < u do
        r := ⌊(l + u)/2⌋ ;
        if F*_{α_r} ∧ ¬φ consistent then
            u := r − 1
        else
            l := r
    {Val(φ, F) = α_r}
end
```

Clearly, this algorithm computes $Val(\phi, \mathcal{F})$ and contains exactly $\lceil \log_2 n \rceil$ calls to SAT. ■

We now focus on *nontrivial deduction problems* in SPL. They are used in order to draw nontrivial, nonmonotonic inference from partially inconsistent possibilistic knowledge bases [Dubois and Prade, 1991a; Dubois *et al.*, 1994].

DEFINITION 14 (nontrivial deduction problem, optimization form). SPL-NTOPT: compute $Val_{NT}(\phi, \mathcal{F}) = \begin{cases} Val(\phi, \mathcal{F}) \text{ iff } Val(\phi, \mathcal{F}) > Incons(\mathcal{F}) \\ 0 \quad\quad\quad\quad \text{otherwise} \end{cases}$

DEFINITION 15 (nontrivial deduction problem, decision form).
$\mathcal{F} \models\!\approx (\phi\ \alpha)$ iff $\mathcal{F} \models (\phi\ \alpha)$ and $\alpha > Incons(\mathcal{F})$
(SPL-NTDED): decide whether $\mathcal{F} \models\!\approx (\phi\ \alpha\}$

DEFINITION 16 (nonmonotonic possibilistic entailment).
(NMPE): decide whether $Val_{NT}(\phi, \mathcal{F}) > 0$.

PROPOSITION 17. *Computing $Val_{NT}(\phi, \mathcal{F})$ requires $1 + \lceil \log_2 n \rceil$ satisfiability checks, where n is the number of different valuations involved in $\mathcal{F}$.*

Proof: immediate corollary of Proposition 11.

PROPOSITION 18. SPL-NTDED *is DP-complete*[2].

Proof: (Membership) : $\mathcal{F} \models\!\approx (\phi\ \alpha)$ iff $\mathcal{F} \models (\phi\ \alpha)$ and $\alpha > Incons(\mathcal{F})$, *i.e.*, iff $\mathcal{F}_\alpha \models \phi$ and $\mathcal{F}_\alpha$ is satisfiable. This requires one satisfiability test and one unsatisfiability test, hence the membership of SPL-NTDED to DP.
(Completeness) : proved by the following polynomial reduction from the canonical DP-complete problem SAT-UNSAT to SPL-NTDED. It is easy to see that (ϕ, ψ) is an instance of SAT-UNSAT, *i.e.*, ϕ is satisfiable and ψ is not, if and only if $\{(\phi\ 1)\} \models\!\approx (\psi \rightarrow \neg\phi\ 1)$: indeed, $\{(\phi\ 1)\} \models\!\approx (\psi \rightarrow \neg\phi\ 1)$ if and only if $\{(\phi\ 1)\} \models (\psi \rightarrow \neg\phi\ 1)$ and $Incons(\{(\phi\ 1)\}) < 1$, *i.e.*; the first condition is equivalent to $\phi \models \psi \rightarrow \neg\phi$ or equivalently, to $\models \neg\psi$; the second one to ϕ satisfiable. $\Diamond$

PROPOSITION 19. *Provided that NP $\neq$ coNP,* NMPE *is in $\Delta_2^P[\mathcal{O}(\ln n)] \setminus$ (NP $\cup$ coNP)*

Proof. (Membership to Δ_2^P) : By Proposition 8, $Val_{NT}(\phi, \mathcal{F}) > 0$ iff $Incons(\mathcal{F} \cup (\neg\phi\ 1)) > Incons(\mathcal{F})$. This can be checked by the following algorithm:
- compute $\alpha = Incons(\mathcal{F} \cup (\neg\phi\ 1))$;
- verify that $\mathcal{F}_\alpha$ is satisfiable

which proves membership of NMPE to $\Delta_2^P[\mathcal{O}(\log n)]$.
(Non-membership to NP) : suppose that NMPE $\in$ NP. Let $\mathcal{F} = \emptyset$, Then checking that ϕ is unsatisfiable is equivalent to $\models\!\approx_{\mathcal{F}} \phi$, which would imply that UNSAT $\in$ NP, *i.e.*, NP = coNP – which is quite unlikely. Non-membership to coNP is showed by similar arguments. ∎

Hence, all forms of deduction problems come down to a small number (logarithmic in the worst case) of calls to the satisfiability problem in classical logic. Hence, there is no real gap of complexity when switching from classical logic to necessity-valued logic. These simple transformations of a possibilistic deduction problem into classical deduction problems enable the construction of simple algorithms for possibilistic logic, directly based on classical theorem provers (see also subsection 3.4). Note that it is not hard to see that if a SPL knowledge base and the formula to prove involve only formulae of a polynomial fragment of propositional logic, then *all* the considered versions of the possibilistic deduction problem become polynomial. Furthermore, in the first-order case, the possibilistic deduction

[2] see in Appendix an introduction to the complexity class DP.

problems have the same computational nature then the associated classical deduction problems (semi-decidable in the general case, decidable iff the associated fragment is decidable).

Note that the complexity results of this section did not need the assumption that formulas be written under conjunctive normal form. In order to extend *resolution* to possibilistic logic, in the next section we will consider clausal forms for possibilistic knowledge bases.

3.2 Clausal form in Standard Possibilistic Logic

A *possibilistic clause* is a possibilistic formula $(c\ \alpha)$ where c is a (propositional or first-order) clause. A *possibilistic clausal form* is a universally quantified conjunction of possibilistic clauses.

The problem of finding a clausal form of $\mathcal{F}$ whose inconsistency degree is the same as $\mathcal{F}$ always has a solution for SPL knowledge bases. Indeed, there exists a clausal form $\mathcal{C}$ of $\mathcal{F}$ such that $Incons(\mathcal{C}) = Incons(\mathcal{F})$, which generalizes the result holding in classical logic about the equivalence between the inconsistency of a set of formulae and the inconsistency of its clausal form. A possibilistic clausal form of a SPL knowledge base can be obtained by the following method, where $\mathcal{F} = \{(\varphi_i\ \alpha_i), i = 1...n\}$

1. put each φ_i in clausal form, i.e. $\varphi_i = (\forall) \wedge_j (c_{ij})$, where c_{ij} is a universally-quantified clause;
2. $\mathcal{C} := (\forall) \wedge_j (c_{ij}\ \alpha_i)$

PROPOSITION 20. *[Dubois* et al.*, 1994]*

$$Incons(\mathcal{C}) = Incons(\mathcal{F})$$

Proof. Comes easily from the equivalence between a classical propositional formula and its clausal form and Proposition 6. ■

3.3 Resolution in Standard Possibilistic Logic

Once a clausal form is defined for a given SPL knowledge base, the resolution principle may easily be extended from classical first-order logic to SPL, in order to compute its inconsistency degree.

The following possibilistic resolution rule between two possibilistic clauses $(c_1\ \alpha_1)$ and $(c_2\ \alpha_2)$ was first proposed in [Dubois and Prade, 1987]:

$$(R) \quad \frac{(c_1\ \alpha_1)\ \ (c_2\ \alpha_2)}{(r(c_1, c_2)\ \min(\alpha_1, \alpha_2))}$$

where $r(c_1, c_2)$ is any classical resolvent of c_1 and c_2.

Besides we introduce the following *subsumption rule*:

$$(S) \quad \frac{(c\ \alpha)}{(c\ \beta) \text{ where } \beta \leq \alpha}$$

If $\mathcal{C}$ is a set of possibilistic clauses, we note $\mathcal{C} \vdash_{Res} (c\ \alpha)$ if $(c\ \alpha)$ can be obtained by a finite number of applications of rules (R) and (S) to $\mathcal{C}$. The following result establishes the soundness of the resolution and subsumption rules:

PROPOSITION 21 (soundness). *[Dubois* et al.*, 1994] If* $\mathcal{C} \vdash_{Res} (c\ \alpha)$ *then* $\mathcal{C} \models (c\ \alpha)$

Proof. If $(r\ \min(\alpha, \beta))$ is obtained by rule (R) from $(c\ \alpha)$ and $(c'\ \beta)$ then $N(r) \geq N(c \wedge c')$ (since $c \wedge c' \models r$), thus $N(r) \geq min(N(c), N(c')) \geq min(\alpha, \beta)$, so $(c\ \alpha), (c'\ \beta) \models (r\ \min(\alpha, \beta))$. Now, if $(c\ \beta)$ is obtained by rule (S) from $(c\ \alpha)$ then trivially $(c\ \alpha) \models (c\ \beta)$. By induction on the derivations, any possibilistic clause obtained by a finite number applications of rules (R) and (S) is a logical consequence of the initial set of possibilistic clauses. ∎

The resolution rule for SPL clauses performs locally at the syntactical level what the combination/projection principle [Zadeh, 1978] does in approximate reasoning.

Moreover, resolution for SPL clauses is complete for refutation:

PROPOSITION 22. *[Dubois* et al.*, 1994] Let* $\mathcal{C}$ *be a set of SPL clauses. Then the valuation of the optimal refutation by resolution from* $\mathcal{C}$ *(i.e., the derivation of* $(\perp\ \bar{\alpha})$ *with* $\bar{\alpha}$ *maximal) is the inconsistency degree of* $\mathcal{C}$.

The proof comes easily from completeness of resolution for refutation in classical first-order logic, together with Propositions 6 and 9. Using Propositions 8 and 10, we get the following corollary:

PROPOSITION 23. *[Dubois* et al.*, 1994] Let* $\mathcal{F}$ *be a set of SPL formulae and* φ *be a classical formula. Let* $\mathcal{C}'$ *be the set of SPL clauses obtained from* $\mathcal{F} \cup \{(\neg\varphi\ 1)\}$*; then the valuation of the optimal refutation by resolution from* $\mathcal{C}'$ *is* $Val(\varphi, \mathcal{F})$.

Thus refutation by resolution can be used for computing the inconsistency degree of a SPL knowledge base. We consider a set $\mathcal{F}$ of SPL formulae (the knowledge base) and a formula φ; we wish to know the maximal valuation with which $\mathcal{F}$ entails φ, i.e. $Val(\varphi, \mathcal{F}) = sup\{\alpha \in (0, 1] | \mathcal{F} \models (\varphi\ \alpha)\}$. This request can be answered by using refutation by resolution, which is extended to standard possibilistic logic as follows:

`Refutation by resolution`:

1. put $\mathcal{F}$ into clausal form $\mathcal{C}$;
2. put φ into clausal form; let c_1, ..., c_m the clauses obtained;

3. $\mathcal{C}' := \mathcal{C} \cup \{(c_1\ 1), ..., (c_m\ 1)\}$;
4. search for a deduction of $(\bot\ \alpha^*)$ by applying the resolution rule (R) from $\mathcal{C}'$ repeatedly, with α^* maximal;
5. $Val(\varphi, \mathcal{F}) := \alpha^*$

Illustrative example

Let $\mathcal{F}$ be the following possibilistic knowledge base, concerning an election whose two only candidates are Mary and Peter:

Φ_1	$(Elected(Peter) \leftrightarrow \neg Elected(Mary)\ 1)$
Φ_2	$(\forall x\ CurrentPresident(x) \to Elected(x)\ 0.5)$
Φ_3	$(CurrentPresident(Mary)\ 1)$
Φ_4	$(\forall x\ Supports(John, x) \to Elected(x)\ 0.6)$
Φ_5	$(Supports(John, Mary)\ 0.2)$
Φ_6	$(\forall x\ VictimOfAnAffair(x) \to \neg Elected(x)\ 0.7)$

$\mathcal{F}$ is equivalent to the set of possibilistic clauses $\mathcal{C} = \{C_1, ..., C_7\}$:

C_1	$(Elected(Peter) \vee Elected(Mary)\ 1)$
C_2	$(\neg Elected(Peter) \vee \neg Elected(Mary)\ 1)$
C_3	$(\neg CurrentPresident(x) \vee Elected(x)\ 0.5)$
C_4	$(CurrentPresident(Mary)\ 1)$
C_5	$(\neg Supports(John, x) \vee Elected(x)\ 0.6)$
C_6	$(Supports(John, Mary)\ 0.2)$
C_7	$(\neg VictimOfAnAffair(x) \vee \neg Elected(x)\ 0.7)$

We cannot find any refutation from $\mathcal{C}$; hence, $\mathcal{C}$ is consistent, i.e., $Incons(\mathcal{C}) = 0$. Let us now find the best lower bound of the necessity degree of the formula $Elected(Mary)$. Let $\mathcal{C}' = \mathcal{C} \cup \{\neg Elected(Mary)\ 1)\}$; then there exist two distinct refutations by resolution from $\mathcal{C}'$, which are the following:

Refutation 1:
$(\neg Elected(Mary)\ 1), C_3 \vdash_{Res} (\neg CurrentPresident(Mary)\ 0.5)$;
$(\neg CurrentPresident(Mary)\ 0.5), C_4 \vdash_{Res} (\bot\ 0.5)$

Refutation 2:
$(\neg Elected(Mary)\ 1), C_5 \vdash_{Res} (\neg Supports(John, Mary)\ 0.6)$;
$(\neg Supports(John, Mary)\ 0.6), C_6 \vdash_{Res} (\bot\ 0.2)$

It can be checked that there is no other refutation from $\mathcal{C}'$. Refutation 1 is optimal, whereas refutation 2 is not. Hence, we conclude that $\mathcal{C} \models (Elected(Mary)\ 0.5)$, i.e., it is moderately certain that Mary will be elected; since this degree 0.5 is maximal, $Val(Elected(Mary), \mathcal{C}) = 0.5$.

Then we learn that Mary is victim of an affair, which leads us to update the knowledge base by adding to $\mathcal{C}$ the possibilistic clause
C_8 : $(VictimOfAnAffair(Mary)\ 1)$. Let $\mathcal{C}_1$ be the new knowledge base, $\mathcal{C}_1 = \mathcal{C} \cup \{C_8\}$. Then we can find a 0.5-refutation from $\mathcal{C}_1$ (which is optimal):
$C_8, C_7 \vdash_{Res} (\neg Elected(Mary)\ 0.7)$;
$C_3, C_4 \vdash_{Res} (Elected(Mary)\ 0.5)$;
$(\neg Elected(Mary)\ 0.7), (Elected(Mary)\ 0.5) \vdash_{Res} (\perp\ 0.5)$

Hence, $\mathcal{C}_1$ is partially inconsistent, with $Incons(\mathcal{C}_1) = 0.5$.

Refutation 1, which had given $N(Elected(Mary)) \geq 0.5$ can still be obtained from $\mathcal{C}_1$ but since its valuation is not greater than $Incons(\mathcal{C}_1)$, it becomes a trivial deduction. On the contrary, adding to $\mathcal{C}_1$ the possibilistic clause $(Elected(Mary)\ 1)$, we find a 0.7-refutation this time:
$(Elected(Mary)\ 1), C_8 \vdash_{Res} (\neg VictimOfAnAffair(Mary)\ 0.7)$;
$(\neg VictimOfAnAffair(Mary)\ 0.7), C_8 \vdash_{Res} (\perp\ 0.7)$

Since $0.7 > Incons(\mathcal{C}_1)$, the deduction $\mathcal{C}_1 \models (\neg Elected(Mary)\ 0.7)$ is non-trivial; using C_1, it could be shown that we also have $\mathcal{C}_1 \models (Elected(Peter)\ 0.7)$.

3.4 Resolution strategies

We have shown that the key problem for possibilistic deduction consists in computing an inconsistency degree, which can be done by searching for an optimal refutation by resolution from a set of possibilistic clauses. In this section we address the issue of finding an optimal refutation *in practice*. As in classical logic, a resolution strategy will restrict or order the choice of clauses or literals on which resolution will be made. In the whole section, $\mathcal{C}$ will denote a set of first-order necessity-valued clauses. $\mathcal{C}^*$ will denote the projection of $\mathcal{C}$.

DEFINITION 24. A possibilistic resolution strategy is an algorithm (deterministic or not) for finding an optimal refutation of $\mathcal{C}$. A possibilistic resolution strategy $\mathcal{S}$ is

- complete iff for any $\mathcal{C}$ such that $Incons(\mathcal{C}) > 0$ there is an optimal refutation for $\mathcal{C}$ compatible with the criteria of $\mathcal{S}$.
- directly complete iff its application to $\mathcal{C}$ such that $Incons(\mathcal{C}) > 0$ leads necessarily to an optimal refutation in a finite amount of time.
- really directly complete iff its application to $\mathcal{C}$ such that $Incons(\mathcal{C}) > 0$ not only leads to an optimal refutation in a finite amount of time, but also recognizes it as optimal.
- decidable iff it is directly complete and furthermore stops after a finite amount of time even if $\mathcal{C}$ is consistent.

A meta-level resolution strategy consists in a meta-level algorithm calling a classical resolution strategy S.

Bottom-up meta-level strategies

These strategies consist in applying a given classical resolution strategy S to increasing α-cuts of $\mathcal{C}$ until the attainment of a consistent α-cut. If the algorithm stops then the last refutation obtained is optimal.

```
begin
  α := 0;
  stop := false;
  repeat
    apply S to C_ᾱ (without caring about valuations);
    if S stops without finding a refutation then
      stop := true
    else
      let β be the valuation of the obtained refutation;
      α := β;
  until stop { possibly given by C = ∅ } ;
  Return α*;
end
```

Let $\mathcal{S}_\uparrow$ be a bottom-up meta-level strategy induced by the classical resolution strategy S.

PROPOSITION 25. *If $\mathcal{S}_\uparrow$ applied to $\mathcal{C}$ stops then the last value of α^* is $Incons(\mathcal{C})$.*

Proof: comes easily from soundness of resolution and proposition.

PROPOSITION 26. *$\mathcal{S}_\uparrow$ is complete (resp. directly complete, decidable) iff S is complete (resp. directly complete, decidable).*

The proof does not contain any particular difficulty [Lang, 1991a]. Note that even if S is directly complete, $\mathcal{S}_\uparrow$ *may not be really directly complete* since the optimal refutation, once found, will be recognized as optimal when the consistency of the remaining set of clauses $\mathcal{C}_{\bar{\alpha}^*}$ is checked, which is not guaranteed to succeed in a finite amount of time.

Top-down meta-level strategies

These strategies consist in applying a given classical resolution strategy S to decreasing α-cuts of $\mathcal{C}$ until the attainment of a refutation (the first one found being optimal). Let us rank the distinct valuations appearing in $\mathcal{C}$ (there are finitely many distinct valuations since $\mathcal{C}$ is finite), namely, $0 < \alpha_1 < ... < \alpha_n \leq 1$.

```
begin
  i := n; stop := false;
  repeat
    apply S to C_{\overline{α_i}} (without caring about valuations) ;
    if a refutation is found then
      α* := α_i;
      stop := true
    else
      i := i − 1
  until stop or i = 0;;
  Return α_i;
end
```

Top-down strategies were also considered by Williams [1997] in the context of belief revision, where an anytime algorithm using a top-down strategy is proposed for computing a particular revision scheme (called maxi-adjustment).

Let $\mathcal{S}_\downarrow$ be a top-down meta-level strategy induced by the classical resolution strategy S.

PROPOSITION 27.
If $\mathcal{S}_\downarrow$ applied to $\mathcal{C}$ stops then the returned value α_i is $Incons(\mathcal{C})$.

Proof. If the last value of i is 0 then no refutation could be found from $\mathcal{C}_{\alpha_1} = \mathcal{C}$, which means that $\mathcal{C}^*$ is consistent (due to the completeness of resolution for refutation), hence $Incons(\mathcal{C}) = 0$ (by Proposition 6). If $i > 0$ then, from the soundness of resolution we get $\mathcal{C}^*_{\alpha_i}$ inconsistent, and thus $Incons(\mathcal{C}) \geq \alpha_i$ by Proposition 6; now, since no refutation could be found from $\mathcal{C}^*_{\alpha_{i+1}}$, from the completeness of resolution for refutation we get that $\mathcal{C}^*_{\alpha_{i+1}}$ is consistent, and again by Proposition 6, $Incons(\mathcal{C}) \leq \alpha_i$. Hence, $Incons(\mathcal{C}) = \alpha_i$. ∎

PROPOSITION 28. *$\mathcal{S}_\downarrow$ is decidable iff S is decidable.*

This result is easy to establish. Note that if S is not decidable (even if it is directly complete), $\mathcal{S}_\downarrow$ is not guaranteed to be complete (and a fortiori directly complete). Indeed, if $Incons(\mathcal{C}) < \alpha_n$, then $\mathcal{C}^*_{\alpha_n}$ is consistent and nothing guarantees that S applied to $\mathcal{C}^*_{\alpha_n}$ will terminate, thus the algorithm may loop for ever in the first iteration.

The search for a refutation of $\mathcal{C}^*_{\alpha_i}$ may take into account the fact that there is no possible refutation from $\mathcal{C}^*_{\alpha_{i+1}}$, and hence, that if there exists a refutation from $\mathcal{C}^*_{\alpha_i}$ then it involves at least a clause with valuation α_i. One possibility is to choose for S a *support set strategy* [Loveland, 1978] with $\mathcal{C}^*_{\alpha_i} \setminus \mathcal{C}^*_{\alpha_{i+1}}$ as support set. Another possibility [Froidevaux and Mengin,] consists in saturating each $\mathcal{C}^*_{\alpha_i}$ by producing all clauses of valuation α_i before trying to produce any clause of $\mathcal{C}^*_{\alpha_{i-1}}$.

Mixed meta-level strategies

Top-down (resp. bottom-up) meta-level strategies are all the more efficient as the inconsistency degree of $\mathcal{C}$ is high (resp. low); hence, the choice of one meta-level strategy or another may be guided by expectations about the inconsistency degree. However, without any prior knowledge about it, applying blindly a top-down or a bottom-up strategy may be very inefficient (in the propositional case, where deduction is decidable, it may lead to n iterations before finding the optimal refutation). In this case, hybrid meta-level strategies may be considered, for instance:

- combining both top-down and bottom-up strategies, e.g., start by searching for a refutation in a high α-cut and in case it does not give anything within a fixed amount of time, try by a bottom-up strategy;

- computing the inconsistency degree by dichotomy (as in the proof of Proposition 11), with at most a logarithmic number of classical resolution proofs.

Informed strategies

The meta-level strategies discussed so far have one drawback: they are blind, in the sense that in a given iteration the choice of the next clauses to resolve is not guided by the valuations. We now briefly define a class of informed strategies, namely, linear pseudo-A^* strategies. The definition of a *linear* resolution strategy for possibilistic logic is exactly the same as in classical logic [Loveland, 1978]: let $\mathcal{C}$ be a set of possibilistic clauses such that $Incons(\mathcal{C}) > 0$, and let us distinguish a clause C_0 in $\mathcal{C}$ (generally chosen such that $\mathcal{C} \setminus \{C_0\}$ be consistent); C_0 is called the initial central clause; then, a linear strategy only admits resolutions between a descendent clause of C_0 (called central clause) and a clause which is either a clause of $\mathcal{C}$ (entry clause) or a descendent of C_0. A linear refutation is a refutation by resolution obtained by the application of a linear strategy.

PROPOSITION 29. *[Lang, 1991a] [Dubois* et al.*, 1987]*
Let $\mathcal{C}$ be a set of possibilistic clauses such that $Incons(\mathcal{C}) > 0$; then, among the optimal refutations of $\mathcal{C}$ there is at least one which is linear.

However, this result does not mean that if an arbitrary clause C_0 is chosen in $\mathcal{C}$ as a central initial clause, an optimal linear refutation from $\mathcal{C}$ with C_0 as central initial clause is an optimal refutation from $\mathcal{C}$. For this to hold, $Incons(\mathcal{C} \setminus \{C_0\}) = 0$. This last condition is guaranteed in case $\mathcal{C}$ was obtained by adding to a consistent possibilistic knowledge base $\mathcal{F}$ one clause corresponding to the negation of the the formula φ to prove: then this clause is chosen as the central initial clause. If the negation of φ gives more than one clause, then it is possible to take successively each one of these clauses, and then retain the maximum of the valuations of obtained optimal refutations[3].

[3]More generally, if one wishes to compute the inconsistency degree of a set of possibilistic clauses which does not come from the addition to the negation of a given formula to a consistent possibilistic knowledge base, one may replace each clause $(c\ \alpha)$ of $\mathcal{C}$ by $(\neg aux \vee c\ \alpha)$ where aux is an auxiliary nullary predicate, and take as central initial clause the new clause $(aux\ 1)$. Thus, in all cases the

The search for a linear refutation by resolution may be expressed in terms of an arborescent search in a state space. A state is defined as a central clause and the set of its ancestor central clauses.

To each state $(C_1...C_n)$ of the search tree is associated the valuation of the last clause obtained $val(C_n)$. Thus we are looking for an objective state with a maximal valuation, where an objective (goal) state has the form $(C_1...(\perp\ \alpha))$ ($\perp$ being the empty clause). Thus, the search for a maximal refutation can be cast as the search for an optimal objective state with min-max costs [Dubois *et al.*, 1987; Lang, 1991a]. Namely:

- the initial state S_0 is defined by the central initial clause C_0 and its cost is $g(S_0) = val(C_0)$;

- the cost associated with an arc $(C_0...C_i) \longrightarrow (C_0...C_iC_{i+1})$ is the valuation of the side clause C'_{i+1} used for producing C_{i+1}.

- the global cost of a path $(C_0) \longrightarrow (C_0C_1) \longrightarrow ... \longrightarrow (C_0...C_{i+1})$ is the minimum of all its elementary costs, which can also be written $g(C_0...C_{i+1}) = min(g(C_0...C_i), val(C'_{i+1}))$;

- an objective (or goal) state is a state $(C_0...C_i)$ such that $C_i = \perp$ ($\perp$ being the empty clause);

- a state $(C_0...C_i)$ is expanded by producing all the resolvents of C_i authorized by the linear strategy.

The search for the empty clause with the highest valuation is equivalent to the search for a path whose cost is maximum. Such state spaces where costs are to be maximized and are combined by $\min$ (instead of $+$) have been studied by Yager [Yager, 1986] who has proposed in this framework (called "possibilistic production systems") a generalisation of algorithms A, A^*, and in a more general framework by Farreny [Farreny, 1997]. Paths with a maximal valuation are called "paths of least resistance". The search is guided by an evaluation function f associating to each state S a valuation $f(S) \in [0,1]$. As for classical A and A^* algorithms, f is a function of two other functions g and h; g is the cost of the path from the initial state S_0 to the current state S, and h is a heuristic function estimating the expected cost from S to an optimal objective state, and

$$f(S) = min(g(S), h(S))$$

(in contrast with usual A and A^* algorithms where $f = g + h$). The next state to expand will be selected among the states maximizing f. We first give the general formulation of the search algorithm, which is independent of the choice of a particular evaluation function.

inconsistency degree of $\mathcal{C}$ may be computed with a linear resolution strategy.

```
begin
  Open := {S_0};
  Success := false;
  α* := 0;
  repeat
    MaxOpen := {S ∈ Open maximizing f};
    if MaxOpen contains an objective state then
      Success := true;
      α := g(S);
    else
      select a state S in MaxOpen;
      remove S from Open;
      develop S; let E_n be the set of newly produced states ;
      compute f for each newly produced state;
      Open := Open ∪E_n;
  until Open = ∅ or Success;
  Return α* {α = Incons(C)}
end
```

A variant of this algorithm consists of maintaining a list `Closed` of states in which all states are placed as they are developed; then, when producing a new state whose last produced clause is C_{n+1}, it is checked whether C_{n+1} is subsumed by a clause ending a state in `Closed`. However it is known that subsumption tests may be too costly with respect to their benefits.

The heuristic function h is said *admissible* iff $\forall S, h(S) \geq h^*(S)$, where $h^*(S)$ is the real cost of an optimal path from state S to an objective state. An admissible function h is thus optimistic (since costs are here to be maximized), like evaluation finctions for traditional A^* algorithms. Note that an example of admissible function is given by $h(S) = 1, \forall S$: in this case we recover the ascending meta-level strategy restricted to linear refutations (corresponding to the uniform-cost strategy in traditional heuristic search). See Farreny [Farreny, 1997] for an extensive study of heuristic search algorithms with non-additive cost functions.

An admissible function h_1 is said to be *more informed than* an admissible function h_2 iff $\forall S, h_1(S) \leq h_2(S)$. Obviously, $h(S) = 1$ corresponds to the least informed admissible function.

PROPOSITION 30. *Assume that the clauses involved in $\mathcal{C}$ are propositional or from a decidable fragment of first-order logic. If furthermore the evaluation function h is admissible, then the above algorithm stops, and the returned value α^* is equal to $Incons(\mathcal{C})$.*

If it is not assumed that clauses are taken from a decidable fragment, then it is no longer guaranteed that the algorithm stops; as an example, consider the evaluation function $h(S) = 1, \forall S$ and $\mathcal{C} = \{(P(x) \rightarrow P(f(x))\ 1),\ (P(a)\ 1), (Q(b)\ 1),$ $(\neg Q(b)\ 0.7)\}$ then the application of the above algorithm never stops (producing $(P(f(a))\ 1)$, $(P(f(f(a)))\ 1)$, etc.), and will never find the 0.7-refutation from $(Q(b)\ 1)$ and $(\neg Q(b)\ 1)$.

However, in the case where the above assumption is not made, we get the following weaker result:

PROPOSITION 31. *Let h be an admissible evaluation function. If the above algorithm stops, then the returned value α^* is equal to $Incons(\mathcal{C})$.*

We also get a result concerning the use of two admissible evaluation functions, one of which is more informed than the other.

PROPOSITION 32. *Let h_1, h_2 be two admissible functions with $h_1 \leq h_2$ (h_1 more informed than h_2). Then any clause produced by the application of the algorithm to $\mathcal{C}$ using h_1 will also be produced by its application to $\mathcal{C}$ using h_2.*

We end this section by proposing a collection of admissible evaluation functions.

Let $\mathcal{C}$ be a set of possibilistic clauses and $R(\mathcal{C})$ the set of possibilistic clauses that can be produced from $\mathcal{C}$ within one application of the resolution principle. Let $R^1(\mathcal{C}) = R(\mathcal{C})$ and $\forall i, R^{i+1}(\mathcal{C}) = R(R^i(\mathcal{C}))$; lastly, let $R^\infty(\mathcal{C}) = \cup_i R^i(\mathcal{C})$. $R^\infty(\mathcal{C})$ contains all clauses producible within any finite resolution steps from $\mathcal{C}$. Now, for any classical first-order clause $c = [\neg]L_1(...) \vee [\neg]L_2(...) \vee ... \vee [\neg]L_n(...)$ we let $Prop(c) = l_1 \vee l_2 \vee ... \vee l_n$, where $l_i = [\neg]L_i(...)$. Thus, $Prop(c)$ is the "propositional projection" of the first-order clause c (obtained by ignoring the arguments if the predicates). Obviously, if c is propositional then $Prop(c) = c$. Lastly, we say that $l \in Prop(c)$ iff l is one of the disjuncts of $Prop(c)$.

Now, let us define

$$H_1(l) = \max\{\alpha, (c\ \alpha) \in R^\infty(\mathcal{C}), \neg l \in c\}$$

What we wish to get is an optimal refutation from $\mathcal{C}$. Let us consider a refutation from $\mathcal{C}$ and let $(c\ \alpha)$ be a clause used in this refutation; for any literal l of $Prop(c)$, the refutation has to use, at one point or the other, a clause $(c'\ \beta)$ such that $Prop(c')$ contains $\neg l$. Therefore, for any $l \in Prop(c)$, any refutation from $\mathcal{C}$ using c necessarily has a valuation not higher than $H_1(l)$. This means, denoting $h(C)$ for $h(S)$ with S being a state whose last clause is C, that

$$\forall C = (c\ \alpha) \in \mathcal{C}, \forall l \in Prop(c), h^*(C) \leq H_1(l)$$

Defined as such, the heuristic function H_1 seems not to be directly computable from $\mathcal{C}$. However, H_1 is directly computable from $\mathcal{C}$, since we have the following result:

PROPOSITION 33.

$$H_1(l) = \max\{\alpha, (c\ \alpha) \in \mathcal{C}, \neg l \in Prop(c)\}$$

Hence, H_1 is a static function which can be easily computed once and for all before the algorithm is executed; namely, the complexity of the computation of H_1 is in $\mathcal{O}(|V|.|\mathcal{C}|)$ where $|V|$ is the number of propositional variables and $|\mathcal{C}|$ the number of clauses in $\mathcal{C}$.

Now, for each clause $C = (c\ \alpha)$, let us define

$$h_1(C) = \begin{cases} 1 & \text{if } c = \perp \\ \min\{H_1(l), l \in Prop(c)\} & otherwise \end{cases}$$

Let $C = (c\ \alpha)$; from $H_1(l) \geq h^*(C) \forall l \in Prop(c)$, we get $h_1(C) \geq h^*(C)$, hence the admissibility of h_1.

Now we define the sequence of evaluation functions $(f_p)_{p \geq 0}$ by

$$f_p : R^\infty(\mathcal{C}) \to [0, 1]$$
$$h_0(C) = 1$$

and $\forall p \geq 0$

$$f_p(C) = \min(val(C), h_p(C))$$
$$h_{p+1}(C) = \min_{l \in Prop(c)} \max_{C' = (c'\ \beta) \in \mathcal{C}, \neg l \in Prop(c')} f_p(C')$$

For $p = 1$ we recover h_1; interestingly, $(h_p)_{p \geq 0}$ is a family of more and more informed admissible evaluation functions:

PROPOSITION 34. $\forall p \geq 0, \forall C \in R^\infty(\mathcal{C})$,

$$h^*(C) \leq h_{p+1}(C) \leq h_p(C)$$

The proof is done by induction on p. Moreover, it can be shown that h_p eventually becomes stationary, namely, $\exists p^*$ such that $\forall p \geq p^*,\ h_p = h_{p^*}$. Moreover, $p^* \leq |\mathcal{C}|$. Let us denote $h_\infty = h_{p^*} = \lim_{p \to \infty} h_p$.

Since the functions h_p are more and more informed when p increases, one might think of using h_p for the largest p possible, preferably h_∞. From $p^* \leq |\mathcal{C}|$ we may think that computing h_∞ should not take so long if $|\mathcal{C}|$ is small. However, for $p > 1$, h_p is not a static function like h_1, and thus the computation must be done again at each resolution step, which may induce a limitation in the practical use of $h_p, p > 1$. A possibilistic theorem prover was implemented, using h_1 as heuristic function [Lang, 1991a].

It is worth mentioning that so-called "deletion strategies" [Loveland, 1978] can be superposed on any possibilistic resolution strategy. Deletion strategies consist of deleting from the current set of produced clauses, some clauses which are guaranteed to be useless in finding an optimal refutation. Some examples of clauses that can be deleted are tautologies, clauses containing pure literals[4] and subsumed clauses. The definition of subsumption in possibilistic logic is stronger than in

[4] A pure literal is a literal whose negation does not appear anywhere in the curent set of clauses.

classical logic: a clause $(c\ \alpha)$ is said to subsume a clause $(c'\ \alpha')$ iff c subsumes[5] c' and $\alpha \geq \alpha'$. For instance, $(P(x)\ 0.7)$ subsumes $(P(a) \vee Q(a)\ 0.4)$ but not $(P(a) \vee Q(a)\ 0.8)$. It is not hard to prove that the deletion of tautologies, clauses containing pure literals and subsumed clauses preserves the inconsistency degree.

4 ALGORITHMS FOR MODEL FINDING IN STANDARD PROPOSITIONAL POSSIBILISTIC LOGIC

In this Section we assume that all possibilistic formulae involved are *propositional*.

While resolution is mainly directed towards finding an inconsistency, *semantic evaluation* is directed towards finding a model of a set of clauses, if any. The terminology "semantic evaluation" is originally from Jeannicot, Oxusoff and Rauzy [Jeannicot *et al.*, 1988] and refers to an improvement of the well-known *Davis and Putnam procedure* [Davis and Putnam, 1960]. After recalling the basics of semantic evaluation in classical logic, we will propose its possibilistic version. Since the notion "being a model of" in possibilistic logic becomes gradual, the possibilistic analog of model finding becomes an optimization problem consisting of finding a model ω that maximizes the possibility $\pi_{\mathcal{F}}(\omega)$ that ω is a model of $\mathcal{C}$, i.e., *optimal model finding*. Since semantic evaluation is defined only in the propositional case, so is its possibilistic version.

4.1 Semantic evaluation in classical propositional logic

The Davis and Putnam procedure [Davis and Putnam, 1960] consists of searching for a model of a set of clauses $\mathcal{C}$ by developing a semantic tree for $\mathcal{C}$ [Chang and Lee, 1973]. The formulation of the Davis and Putnam procedure we give here is from [Jeannicot *et al.*, 1988].

Let $\mathcal{C}$ be a set of propositional clauses; let p be a literal appearing in $\mathcal{C}$. We denote by $T_p(\mathcal{C})$ the set of clauses obtained from $\mathcal{C}$ by removing all clauses containing p, and by deleting all occurrences of $\neg p$ in the remaining clauses. For instance, if

$$\mathcal{C} = \{a \vee b, \neg a \vee b, \neg b \vee c, \neg c \vee a\}$$

then

$$T_{\neg a}(\mathcal{C}) = \{b, \neg b \vee c, \neg c\}$$

and

$$T_{\neg a,c}(\mathcal{C}) = T_c(T_{\neg a}(\mathcal{C})) = T_{\neg a}(T_c(\mathcal{C})) = \{b, \bot\}$$

A clause c subsumes a clause c' iff the set of literals of c is included in the set of literals of c' (for instance, $a \vee \neg b$ subsumes $a \vee \neg b \vee c$). For any set of clauses $\mathcal{C}$ we denote by $S(\mathcal{C})$ the simplification by subsumption of $\mathcal{C}$, obtained by removing

[5] c classically subsumes c' iff there exists a substitution σ such that $c_\sigma \subseteq c'$.

from $\mathcal{C}$ all clauses subsumed by another one. Lastly, we let $T_p^*(\mathcal{C}) = S(T_p(\mathcal{C}))$. Then, the following property [Jeannicot *et al.*, 1988]:

$$\begin{aligned} \mathcal{C}\ consistent &\Leftrightarrow T_p(\mathcal{C})\ consistent\ or\ T_{\neg p}(\mathcal{C})\ consistent \\ &\Leftrightarrow T_p^*(\mathcal{C})\ consistent\ or\ T_{\neg p}^*(\mathcal{C})\ consistent \end{aligned}$$

gives us a recursive algorithm for testing the consistency of $\mathcal{C}$:

Function `Consistency` ($\mathcal{C}$):

```
begin
    if C = ∅ then
        Return (True)
    else
        if ⊥ ∈ C then
            Return (False)
        else
            choose a literal p appearing in C;
            Return (Consistency(T*_p(C)) ∨ Consistency (T*_¬p(C)));
end
```

It can be shown [Jeannicot *et al.*, 1988] that the efficiency of the Davis and Putnam procedure can be improved by cutting some branches of the semantic tree, due to the *model partition theorem* that we recall here: let $\mathcal{C}$ be a set of clauses and $p_1, ..., p_j, ..., p_k$ distinct literals appearing in $\mathcal{C}$ and such that $\{p_1, ..., p_k\}$ does not contain any complementary pair (i.e., a literal and its negation); let $\mathcal{C}_j = T^*_{p_1,\ldots,p_j}(\mathcal{C})$ and $\mathcal{C}_k = T^*_{p_1,\ldots,p_j,p_{j+1},\ldots,p_k}(\mathcal{C})$. Then:

if $\mathcal{C}_k \subseteq \mathcal{C}_j$ then the consistency of $\mathcal{C}_k$ is equivalent to the consistency of $\mathcal{C}_j$

Thus, for testing whether $\mathcal{C}_j$ is consistent, it will be enough to test whether $\mathcal{C}_k$ is consistent; this avoids studying the branches located between $\mathcal{C}_j$ and $\mathcal{C}_k$.

We call *semantic evaluation* the algorithm resulting from the addition of the cuts due to the model partition theorem to the standard Davis and Putnam procedure. For some varieties of problems (especially those which are "strongly consistent" in the sense that the set of clauses has a large number of models), semantic evaluation can be much more efficient than resolution. Furthermore it gives a constructive proof of consistency by exhibiting a model. Lastly, semantic evaluation is relatively insensitive to syntax, since it can be applied to non-clausal propositional formulas as well.

4.2 Possibilistic semantic trees

A possibilistic semantic tree is similar to a classical semantic tree, the difference being due to the gradation of inconsistency.

DEFINITION 35. Let $\mathcal{C}$ be a set of propositional possibilistic clauses; let $A = \{a_1, ..., a_n\}$ be the set of propositional atoms appearing in $\mathcal{C}$. A *partial world* assigns a truth value to propositional variables in a subset A' of A; a *(complete) world* assigns a truth value to all propositional variables of A. We denote partial world (including complete worlds which are special cases of partial worlds) by v, v' etc., and complete worlds by ω, ω', etc. D_v denotes the subset of propositional variables assigned a truth value by v. For any two partial worlds v, v' we say that v' extends v (denoted by $v' \sqsupseteq v$) iff $D_{v'} \supseteq D_v$ and $\forall a \in D_v$, v and v' assign the same truth value to a.

We recall that if $\mathcal{C} = \{(c_i\ \alpha_i), i = 1...n\}$ then the degree to which it is possible that ω be the actual world is

$$\pi^*_{\mathcal{C}}(\omega) = \min\{1 - \alpha_i | \omega \models \neg c_i\}$$

We extend $\pi^*_{\mathcal{C}}$ to partial worlds by

$$\pi^*_{\mathcal{C}}(v) = \max\{\pi^*_{\mathcal{C}}(\omega) | \omega \sqsubseteq v\}$$

and we also define

$$inc(\mathcal{C}, \omega) = 1 - \pi^*_{\mathcal{C}}(\omega)$$

$$inc(\mathcal{C}, v) = 1 - \pi^*_{\mathcal{C}}(v)$$

Lastly, for any partial world v we let $For(v)$ be a propositional formula whose models are exactly $\{\omega | \omega \sqsubseteq v\}$ ($For(v)$ is unique up to logical equivalence).

In classical propositional logic, a semantic tree for the set of clauses $\mathcal{C}$ is a binary tree whose nodes correspond to partial worlds, and leaves to complete worlds. At each node, a new propositional variable is chosen among the variables not instanciated yet. In a possibilistic semantic tree, some extra information pertaining to the inconsistency degree is needed. More formally:

DEFINITION 36. A possibilistic semantic tree for $\mathcal{C}$ is a classical semantic tree to each node of which is attached a valuation; each node corresponds to a partial world v (complete if the node is terminal), and the valuation attached to it is $inc(\mathcal{C}, v)$.

The valuations of the terminal nodes can be computed directly; the valuations of non-terminal nodes can then be computed bottom-up, due to the following result.

PROPOSITION 37. *let v be a non-complete partial world and a_i a propositional variable non assigned by v; let v_1 and v_2 the partial worlds obtained from v by assigning a_i respectively to $true$ and $false$. Then*

$$inc(\mathcal{C}, v) = \min(inc(\mathcal{C}, v_1), inc(\mathcal{C}, v_2))$$

Proof. $\{\omega | v \sqsubseteq \omega\} = \{\omega | v_1 \sqsubseteq \omega\} \cup \{\omega | v_2 \sqsubseteq \omega\}$, hence the result. ∎

This property enables us to compute inconsistency degrees recursively; it will be the basis of the possibilistic version of the Davis and Putnam procedure.

The relevant information in a possibilistic semantic tree is the set of terminal nodes with the lowest valuation $inc(\mathcal{C},\omega)$, which are actually the *preferred models* w.r.t. $\mathcal{C}$, and the value associated with the root of the tree (which is the value associated to the preferred models), which is the inconsistency degree of $\mathcal{C}$:

PROPOSITION 38. *Let $\mathcal{T}$ be a semantic tree for $\mathcal{C}$; then the value associated to the root of $\mathcal{T}$ is the inconsistency degree of $\mathcal{C}$.*

Proof. It is based on the following lemma:

LEMMA 39. *For each node of $\mathcal{T}$ corresponding to the partial valuation v,*

$$inc(\mathcal{C},v) = Incons(\mathcal{C} \wedge (For(v)\ 1)))$$

The lemma is now proved by upward induction on the tree.

For each terminal node, associated to a complete world ω, we have $Incons(\mathcal{C} \wedge For(\omega)) = 1 - \max_{\omega' \in \Omega} \min(\pi^*_{\mathcal{C}}(\omega'), \pi^*_{\{(For(\omega)\ 1)\}}(\omega'))$; now, $\pi^*_{\{(For(\omega)\ 1)\}}(\omega') = 0$ if $\omega' \neq \omega$, and $= 1$ if $\omega' = \omega$, hence $Incons(\mathcal{C} \wedge For(\omega)) = 1 - \pi^*_{\mathcal{C}}(\omega) = inc(\mathcal{C},\omega)$.

Now let us prove the property by upward induction; let a non-terminal node associated to a partial world v, and assume the partial worlds v_1 and v_2 associated to its children satisfy the property. Then, $Incons(\mathcal{C} \wedge (For(v)\ 1)) = Incons(\mathcal{C} \wedge (For(v_1) \vee For(v_2)\ 1)) = 1 - \max_{\omega \in \Omega} \min(\pi^*_{\mathcal{C}}(\omega), \pi^*_{\{(For(v_1) \vee For(v_2)\ 1)\}}(\omega)) = 1 - \max_{\omega \models v_1 \vee v_2} \pi^*_{\mathcal{C}}(\omega) = \min(inc(\mathcal{C},v_1), inc(\mathcal{C},v_2)) = inc(\mathcal{C},v)$ by Proposition 29, which achieves proving the lemma.

Now, the application of the lemma to the root of the tree, associated to the "empty world" v_0, gives $inc(\mathcal{C},v_0) = Incons(\mathcal{C} \wedge For(v_0)) = Incons(\mathcal{C} \wedge \top) = Incons(\mathcal{C})$. ■

Example: Let $\mathcal{C} = \{C_1 : (p\ 0.8),\ C_2 : (p \rightarrow r\ 0.4),\ C_3 : (p \rightarrow q\ 0.5),\ C_4 : (\neg q\ 0.2),\ C_5 : (\neg r\ 0.1),\ C_6 : (p \rightarrow \neg r\ 0.3)\}$.

A possibilistic semantic tree for $\mathcal{C}$ is represented in Figure 1:

Let us detail the computation of the value attached to the terminal node corresponding to the complete world $\{p,q,r\}$; since $\{p,q,r\}$ falsifies the possibilistic clauses $(\neg q\ 0.2)$, $(\neg r\ 0.1)$ and $(p \rightarrow \neg r\ 0.3)$, we get $inc(\mathcal{C},\{p,q,r\}) = 1 - \pi^*_{\mathcal{C}}(\{p,q,r\}) = 1 - \min\{1 - \alpha \mid (c\ \alpha) \in \mathcal{C}, \{p,q,r\} \models \neg c\} = \max\{\alpha | (c\ \alpha) \in \mathcal{C}, \{p,q,r\} \models \neg c\} = \max(0.2, 0.1, 0.3) = 0.3$.

The value attached to the root of the tree is 0.3, hence $Incons(\mathcal{C}) = 0.3$. There is only one preferred model of $\mathcal{C}$, namely, $\{p,q,r\}$.

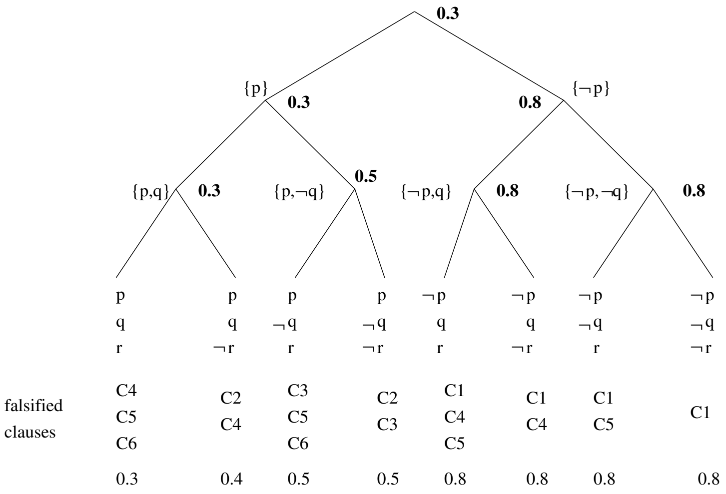

Figure 1.

4.3 *Possibilistic semantic evaluation*

Basic principle

Let $\mathcal{C}$ be a set of propositional possibilistic clauses. $T_p(\mathcal{C})$ and $T_p^*(\mathcal{C})$ are defined as before (the weights being unchanged).

Due to Propositions 37 and 38, $Incons(\mathcal{C})$ can be computed by the following recursive algorithm, which is the basis of the possibilistic version of the Davis and Putnam procedure.

Function `Incons` ($\mathcal{C}$):

begin
  if $\mathcal{C} = \emptyset$ **then**
    Return 0
  else
    if $\mathcal{C} = \{(\bot\ \alpha)\}$ (1) **then**
      Return α
    else
      choose a literal p appearing in $\mathcal{C}$;
      Return (min$(Incons(T_p^*(\mathcal{C})), Incons(T_{\neg p}^*(\mathcal{C})))$)
end

Due to the simplification by subsumption made at each step, it cannot be the case that several empty clauses belong to $\mathcal{C}$ at step [1][6].

The difference with the classical procedure of Davis and Putnam is the absence of a systematic stop when an empty clause is discovered. The above algorithm stops and returns the inconsistency degree of $\mathcal{C}$, and the best model(s) of $\mathcal{C}$ is (are) the worlds corresponding to the nodes giving its weight to the root of the tree.

Note that the possibilistic Davis and Putnam procedure is close to the backtrack plus forward checking algorithm in the framework of fuzzy constraint satisfaction problems ([Dubois *et al.*, 1996]) — up to the difference between the languages of representation.

Representation by minimax trees

The possibilistic procedure of Davis and Putnam, executed such as described above, generally needs visiting most of the 2^n possible branches (where n is the number of propositional variables appearing in $\mathcal{C}$), because the halting criterion is too weak and thus may be rarely fulfilled. However it can be improved so as to take account of partial inconsistencies as soon as possible. Thus, each time an empty clause of weight α appears in $\mathcal{C}$, it will be removed as well as all clauses with a weight $\leq \alpha$, *i.e.*, $\mathcal{C}$ is replaced by $\mathcal{C}_{\bar{\alpha}}$ and $Incons(\mathcal{C}_{\bar{\alpha}})$ is then computed. $Incons(\mathcal{C})$ will be the maximum of α and $Incons(\mathcal{C}_{\bar{\alpha}})$. This is equivalent to replacing each portion of the tree (left part of Figure 2) by the following one (right part of the Figure 2).

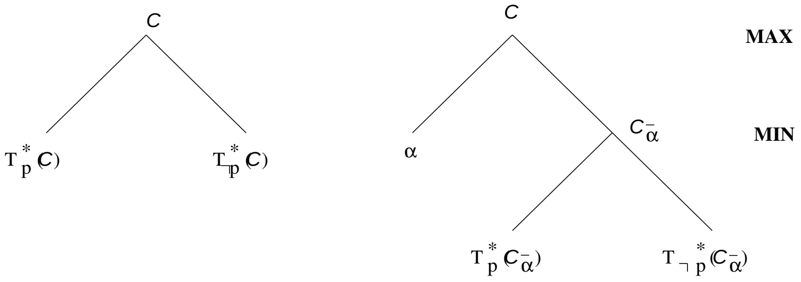

Figure 2.

Thus the possibilistic semantic tree has been translated into an equivalent minimax tree. The new recursive algorithm for computing $Incons(\mathcal{C})$ corresponds to the exploration of this minimax tree:

[6]If the subsumption test is omitted [i.e. if we replace $T^*_p(\mathcal{C})$ and $T^*_{\neg p}(\mathcal{C})$ by $T_p(\mathcal{C})$ and $T_{\neg p}(\mathcal{C})$] then the test *if* $\mathcal{C} = \{(\perp\ \alpha)\}$ *then return* α should be replaced by *if* $\mathcal{C}$ *contains only empty clauses then return the highest value attached to an empty clause in* $\mathcal{C}$.

Function `Incons` ($\mathcal{C}$):

begin
Create the root $\langle n_0, \mathcal{C}\rangle$ and give it the status MIN;
Return (`ValuationMin`($\langle n_0, \mathcal{C}\rangle$))
end

Function `ValuationMin` ($\langle n, \mathcal{C}_n\rangle$):

begin
if $\mathcal{C}_n = \emptyset$ **(1) then**
Return 0
else
Choose a literal p appearing in $\mathcal{C}_n$;
Create a left child $\langle n_1, T_p^*(\mathcal{C}_n)\rangle$, with status MAX;
$v_1 :=$ `ValuationMax` ($\langle n_1, T_p^*(\mathcal{C}_n)\rangle$);
Create a right child $\langle n_2, T_p^*(\mathcal{C}_n)\rangle$, with status MAX;
$v_2 :=$ `ValuationMax` ($\langle n_2, T_p^*(\mathcal{C}_n)\rangle$);
Return ($\min(v_1, v_2)$);
end

Function `ValuationMax` ($\langle n, \mathcal{C}_n\rangle$):

begin
if $\mathcal{C}_n$ *contains an empty clause* $(\perp \alpha)$ **(2) then**
$v_{inc} := \alpha$
else
$v_{inc} := 0$;
create a right child $\langle n', (\mathcal{C}_n)_{\bar{\alpha}})\rangle$;
$v' :=$ ValuationMin ($\langle n', (\mathcal{C}_n)_{\bar{\alpha}})\rangle$);
Return ($\max(v_{inc}, v')$)
end

(1) Note that we have $\mathcal{C}_n = \emptyset$ in only two cases: either all propositional variables from the initial set of clauses have been given a truth value, or the partial interpretation corresponding to the node satisfies all clauses whose valuation is greater than the highest valuation of all empty clauses between the root and the node.
(2) If the algorithm contains at each step a simplification by subsumption of the set of clauses, then at this step of algorithm (2), $\mathcal{C}_n$ contains *at most* one empty clause. If the subsumption check is not made, and if several empty clauses appear at this step then the one with the highest valuation is retained.

PROPOSITION 40. *The function $ValuationMin$ applied to the set of clauses $\mathcal{C}_n$ and the node n computes exactly the inconsistency degree of $\mathcal{C}_n$.*

Proof. By bottom-up induction on the tree and the following lemma (obvious from Proposition 6):
$Incons((\mathcal{C}_n)_{\bar{\alpha}}) = 0$ if $\alpha \geq Incons(\mathcal{C}_n)$ and $Incons(\mathcal{C}_n)_{\bar{\alpha}} = Incons(\mathcal{C}_n)$ if $\alpha < Incons(\mathcal{C}_n)$.
If node n has no child then $\mathcal{C}_n = \emptyset$ and thus $ValuationMin(n, \mathcal{C}_n) = 0 = Incons(\mathcal{C}_n)$, which establishes the base induction step.
Now, assume that all children of $\mathcal{C}_n$ verify the induction hypothesis.

- if n is a Min node then let n_l and n_r its children; then,
  $ValuationMin(n) = \min(ValuationMin(n_l), ValuationMin(n_r))$
  $= \min(Incons(\mathcal{C}_l), Incons(\mathcal{C}_r))$ by the induction hypothesis applied to n_l and n_r
  $= \min(Incons(T_p(\mathcal{C}_n)), Incons(T_{\neg p}(\mathcal{C}_n)))$
  $= Incons(\mathcal{C}_n)$.

- if n is a Max node then, let α the valuation of the highest empty clause appearing in $\mathcal{C}_n$ (or 0 if none appears), and let n_r be its right child, associated to the empty clause free set of clauses $\mathcal{C}_r$. If $\alpha = 0$ then $\mathcal{C}_{n'} = \mathcal{C}_n$ and thus $ValuationMax(n) = \max(0, ValuationMin(n')) = ValuationMin (n') = Incons(\mathcal{C}_{n'}) = Incons(\mathcal{C}_n)$. Now, it remains to prove that if $\alpha > 0$ then $Incons(\mathcal{C}_n) = \max(\alpha, Incons((\mathcal{C}_n)_{\bar{\alpha}})))$. It cannot be the case that $Incons(\mathcal{C}_n) < \alpha$ because $\mathcal{C}_n$ contains $(\bot\ \alpha)$. If $Incons(\mathcal{C}_n) < \alpha$ then the lemma guarantees that $Incons((\mathcal{C}_n)_{\bar{\alpha}}) = 0$ and thus $\max(\alpha, Incons ((\mathcal{C}_n)_{\bar{\alpha}}))) = \alpha = Incons(\mathcal{C}_n)$.
  Lastly, if $Incons(\mathcal{C}_n) > \alpha$ then the lemma tells us that $Incons((\mathcal{C}_n)_{\bar{\alpha}}) = Incons(\mathcal{C}_n)$, hence, $\max(\alpha, Incons((\mathcal{C}_n)_{\bar{\alpha}})) = \max(\alpha, Incons(\mathcal{C}_n)) = Incons(\mathcal{C}_n)$, which achieves proving the proposition. ∎

PROPOSITION 41.
The function $ValuationMin$ applied to the initial set of clauses and the root of the tree computes exactly the inconsistency degree of $\mathcal{C}$.

This is an immediate corollary of Proposition 31.

4.4 *Pruning the tree*

The possibilistic Davis and Putnam procedure can then be improved by cutting a certain number of branches in the corresponding min-max tree. They are two kinds of such cuts:

Alpha-beta cuts

These cuts [Knuth and Moore, 1975] are classical in min-max tree search. To each created node we attach a temporary return value (TRV) which is, for a MAX (resp. MIN) node, the largest (resp. least) of the definitive values of its children; it is thus a lower (resp. upper) bound of its definitive TRV, which becomes equal to the definitive value at the latest when all children nodes have been explored or cut.

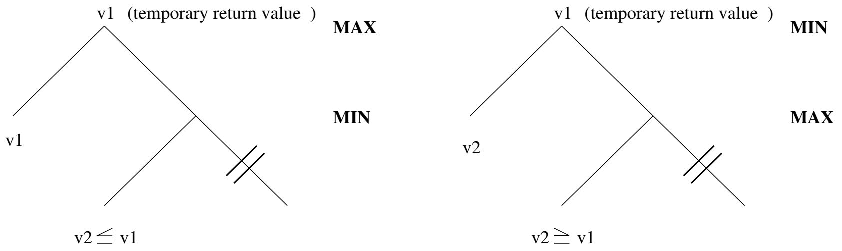

Figure 3.

When computing the value of a MAX node n (Figure 3, left part), assume that the definitive values v_1 and v_2 have been attached to its left child l and to the left child lr of its right child r, and that furthermore we have $v_2 \leq v_1$; then, whatever the value attached to the right child r of n, the value computed upwards for r will in any case be $\leq v_2$ and a fortiori $\leq v_1$, so that the TRV of n becomes definitive, and the right branch issued from n does not have to be explored. The case for computing the value of a MIN node (right part of Figure 3) is symmetrical. More generally, this cutting property still holds as soon as the MIN (resp. MAX) node with the value $v_2 \leq v_1$ (resp. $v_2 \geq v_1$) is a descendent (not necessarily a child) of the MAX (resp. MIN) node n.

Model partition cuts

The model partition theorem ([Jeannicot *et al.*, 1988], recalled in Section 4.1) can be extended to standard possibilistic logic in the following way:

PROPOSITION 42. *Let $\mathcal{C}$ a set of possibilistic clauses; let $p_1, ..., p_k$ be distinct literals appearing in $\mathcal{C}$ and such that $\{p_1, ..., p_k\}$ does not contain any complementary pair. Let $\mathcal{C}_j = T^*_{p_1,...,p_j}(\mathcal{C})$ and $\mathcal{C}_k = T^*_{p_{j+1},...,p_k}(\mathcal{C}_j)$.*
Let β the highest inconsistency value on the way from $\mathcal{C}_j$ to $\mathcal{C}_k$ if there is at least one, with $\beta = 0$ otherwise. Then, if

$$\beta \leq Incons(\mathcal{C}_k) \leq Incons(\mathcal{C}_j)$$

then

$$Incons(\mathcal{C}_k) = Incons(\mathcal{C}_j) \quad (1)$$

and the corresponding cuts can thus be made.

Remark: (1) is often a consequence of one of these two conditions:

- $\mathcal{C}_k \subseteq \mathcal{C}_j$ where $\subseteq$ denotes classical set inclusion;

- more generally, $\mathcal{C}_k \subseteq \mathcal{C}_j$ where $\subseteq$ denotes fuzzy set inclusion, *i.e.*, $\forall (c\ \alpha) \in \mathcal{C}_k\ \exists (c'\ \beta) \in \mathcal{C}_j$ such that $c' = c$ and $\beta \geq \alpha$.

This result generalizes the classical model partition theorem, where the node $\mathcal{C}_k$ is studied only if its ancestors do not contain any empty clause, *i.e.*, if $\beta = 0$.

Proof. (of Proposition 33). Let ω_k an optimal model of $\mathcal{C}_k$, *i.e.*, $inc(\mathcal{C}_k, \omega_k) = Incons(\mathcal{C}_k)$. Let ω_j the interpretation assigning to all literals of $\mathcal{C}_k$ the same truth value as ω_k and furthermore satisfying $p_{j+1}, ..., p_k$. We have

$$\begin{aligned} inc(\mathcal{C}_j, \omega_j) &= Incons(\mathcal{C}_j \wedge For(\omega_j))\ (Lemma 30) \\ &= \max(\beta, inc(\mathcal{C}_k, \omega_k)) \end{aligned}$$

because $T^*_{p_{j+1},\ldots,p_k}(\mathcal{C}_j \wedge For(\omega_j)) = \mathcal{C}_k$ and the optimal interpretation of $\mathcal{C}_j \wedge For(E_{\omega_j})$ satisfies necessarily $p_{j+1}, ..., p_k$. Then, $inc(\mathcal{C}_k, \omega_k) = Incons(\mathcal{C}_k) \geq \beta$ by hypothesis, hence $inc(\mathcal{C}_j, \omega_j) = inc(\mathcal{C}_k, \omega_k) = Incons(\mathcal{C}_k)$, and since $Incons(\mathcal{C}_j) = inf_\omega inc(\mathcal{C}_j, \omega)$ we have $Incons(\mathcal{C}_j) \leq inc(\mathcal{C}_j, \omega_j)$ and thus $Incons(\mathcal{C}_j) \leq Incons(\mathcal{C}_k)$. Since we have by hypothesis $Incons(\mathcal{C}_k) \leq Incons(\mathcal{C}_j)$, we conclude that $Incons(\mathcal{C}_k) = Incons(\mathcal{C}_j)$. ■

Example The initial set of clauses (1) is $\mathcal{C} = \{p\ 0.8), (\neg p \vee r\ 0.4), (\neg p \vee q\ 0.5), (\neg q\ 0.2), (\neg r\ 0.1), (\neg p \vee \neg r\ 0.3)\}$. Here is the corresponding min-max tree (where the order of instanciation is q, p, r in all branches), taking benefit from cuts.

The tree is explored depth-first from left to right (Figure 4). The values are computed bottom-up. A step (2) we see that $T^*_q(\mathcal{C}) \subseteq \mathcal{C}$, thus the model partition theorem can be applied and the branch corresponding to $T^*_q(\mathcal{C})$ can be cut off. At step (3) an alpha-beta cut can be done. The optimal model of $\mathcal{C}$ is pqr; the inconsistency degree of $\mathcal{C}$ is 0.3.

5 ALGORITHMS FOR EXTENSIONS OF POSSIBILISTIC LOGIC

5.1 Extended possibilistic logic

Standard possibilistic logic handles only "certainty-qualified statements". It has been already said that it is practically sufficient for most applications (see [Dubois *et al.*, 1994] and Conclusion for a panorama of applications). However there are some situations where it is also desirable to handle "possibility-qualified statements". These kinds of statements (see [Dubois and Prade, 1987; Dubois *et al.*, 1991a; Dubois *et al.*, 1994]) are represented semantically by constraints of the form $\Pi(\varphi) \geq \alpha$.

Thus, in Extended Possibilistic Logic (EPL), a knowledge base is a set of EPL formulas being either necessity valued formulas of SPL, denoted here by $(\varphi\ (N\ \alpha))$ or possibility-valued formulas $(\varphi\ (\Pi\ \beta))$ with $0 < \beta \leq 1$. A possibility-valued formula $(\varphi\ (\Pi\ \beta))$ intuitively expresses to what extent φ cannot be refuted.

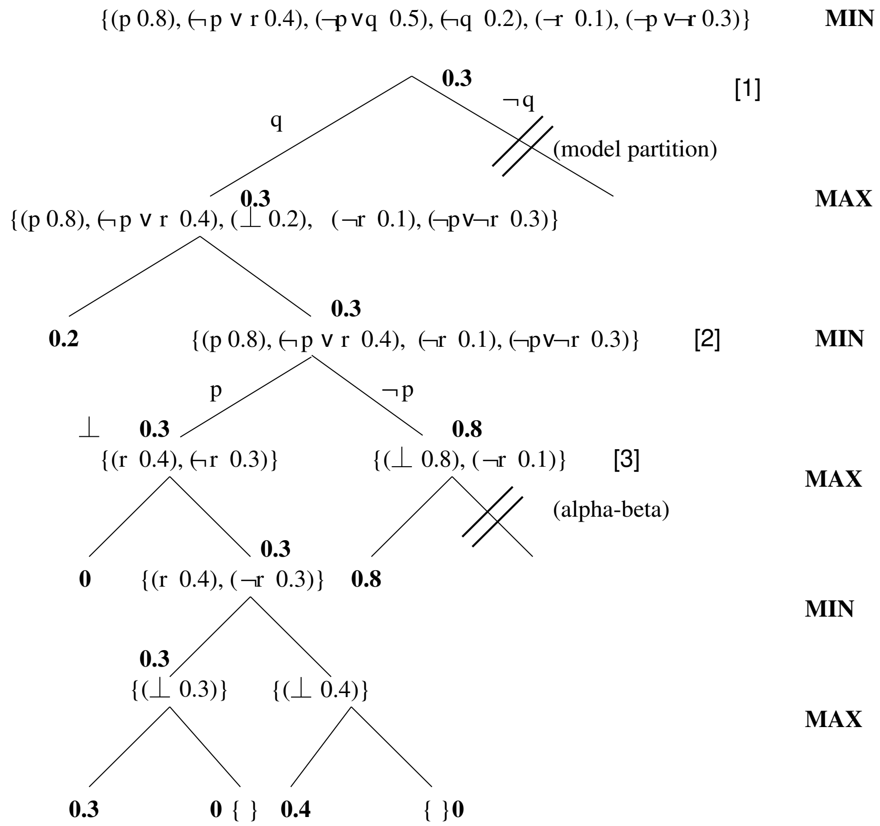

Figure 4.

Note that $\Pi(\varphi) \geq \beta$ is equivalent to $N(\neg\varphi) \leq 1-\beta$, which shows that possibility-valued formulas are complementary to necessity-valued ones, since they express an *upper* bound on the certainty level. EPL more generally enables a larger variety of types of constraints on $N(\varphi)$; for instance, $\alpha \leq N(\varphi) \leq \beta$ can be expressed in EPL by $\{(\varphi\ (N\ \alpha)), (\neg\varphi\ (\Pi\ 1-\beta))\}$. In particular, $\{(\varphi\ (\Pi\ 1)), (\neg\varphi\ (\Pi\ 1))\}$ expresses that $\Pi(\varphi) = \Pi(\neg\varphi) = 1$, or equivalently $N(\varphi) = N(\neg\varphi) = 0$, i.e., expresses *explicitly* complete ignorance about φ, which SPL alone cannot do.

An interpretation of possibilistic logic where possibility-qualified statements are particularly meaningful is when dealing with graded obligations and permissions; while $N(\varphi) \geq \alpha$ expresses that φ is obligatory to the degree α, $\Pi(\varphi) \geq \beta$ expresses that φ is at least permitted to the degree α, which is much weaker.

Since $N(\varphi) > 0$ entails $\Pi(\varphi) = 1$, $(\varphi\ (N\alpha))$ is stronger than $(\varphi\ (\Pi\ \beta)$ for any $\alpha > 0, \beta > 0$. This leads to the following ordering between valuations:

$(N\ \alpha) \leq (N\ \beta)$ if $\alpha \leq \beta$; $(\Pi\ \alpha) \leq (\Pi\ \beta)$ if $\alpha \leq \beta$; $(N\ \alpha) \leq (N\ \beta)\ \forall \alpha, \beta > 0$. We let $\mathcal{W} = \{(N\ \alpha), \alpha > 0\} \cup \{(\Pi\ \beta), \beta > 0\}$ the set of all EPL valuations.

The semantics of EPL generalizes SPL's ([Lang, 1991c; Dubois *et al.*, 1994]); in particular we have
$\pi \models (\varphi\ (\Pi\ \beta))$ iff $\Pi(\varphi) \leq \beta$.
The function Val is extended to EPL by $Val(\varphi, \mathcal{F}) = Sup\{w \mid \mathcal{F} \models (\perp\ w)\}$. This leads to a richer view of grader inconsistency, where $(\Pi\ \beta)$-inconsistencies are weaker than $(N\ \alpha)$-inconsistencies (see [Dubois *et al.*, 1994] for a complete exposition of EPL).

The following results are useful for extending SPL proof techniques to EPL. The first one extends Proposition 8. The second one has been independently proved in [Dubois *et al.*, 1994] and by Hollunder [Hollunder, 1994].

PROPOSITION 43.

- $\mathcal{F} \models (\varphi\ w)$ *iff* $\mathcal{F} \cup \{(\neg\varphi\ (N\ 1))\} \models (\perp\ w)$*;*
- $Val(\varphi, \mathcal{F}) = Incons(\mathcal{F} \cup \{(\varphi\ (N\ 1))\}$

PROPOSITION 44.
Let $\mathcal{F} = \mathcal{F}_N \cup \mathcal{F}_\Pi$ *where* $\mathcal{F}_N$ *and* $\mathcal{F}_\Pi$ *contain respectively the necessity-valued and possibility-valued formulas of* $\mathcal{F}$*. Let* $\mathcal{F}_N^* = \{\varphi \mid (\varphi\ (N\ \alpha)) \in \mathcal{F}\}$*. Then*

- $Incons(\mathcal{F}) = (N\ \alpha)$ *iff* $Incons(\mathcal{F}_N) = (N\ \alpha)$
- $Incons(\mathcal{F}) = (\Pi\ \beta)$ *iff* $\mathcal{F}_N^*$ *is consistent and* $\exists(\varphi\ (\Pi\ \beta)) \in \mathcal{F}_\Pi$ *such that*
  - $\mathcal{F}_N^* \cup \{\varphi\}$ *is inconsistent*
  - $\forall\beta' > \beta$*,* $\forall(\varphi'\ (\Pi\ \beta')) \in \mathcal{F}_\Pi$*,* $\mathcal{F}_N^* \cup \{\varphi'\}$ *is consistent.*

Proposition 34 means that like in SPL, any deduction problem in EPL can be reformulated equivalently into the computation of an inconsistency degree; Proposition 35 gives us an easy way to practically compute $Incons(\mathcal{F})$, together with complexity results for deduction in EPL.

PROPOSITION 45. *Deciding that* $\mathcal{F} \models (\varphi\ w)$ *in EPL is* coNP*-complete.*

PROPOSITION 46. *Computing* $Val(\varphi, \mathcal{F})$ *in EPL requires at most* $|\mathcal{F}_\Pi| + \lceil \log_2 |\mathcal{F}_N| \rceil$ *satisfiability checks.*

Now, resolution can be extended to EPL:

DEFINITION 47 (resolution in EPL). [Dubois and Prade, 1990] The operation $\odot : \mathcal{W} \times \mathcal{W} \to \mathcal{W}$ is defined by

$$\begin{cases} (N\ \alpha) \odot (N\ \beta) & = (N\ \min(\alpha, \beta)) \\ (N\ \alpha) \odot (\Pi\ \beta) & = \begin{cases} (\Pi\ \beta) & \mathrm{i}f \alpha + \beta > 1 \\ (\Pi\ 0) & \mathrm{o}therwise \end{cases} \\ (\Pi\ \alpha) \odot (\Pi\ \beta) & = (\Pi\ 0) \end{cases}$$

The resolution rule for two clauses $(c_1\ w_1)$ and $(c_2\ w_2)$ of EPL is then defined by:

$$\frac{(R)\quad (c_1\ w_1)\ (c_2\ w_2)}{(r(c_1, c_2)\ w_1 \odot w_2)}$$

PROPOSITION 48. *(R) is sound*

As to completeness, the result is not so easy as in SPL; in order to generalize Proposition 18 we must assume that $\mathcal{F}$ is *already in clausal form*[7] and *propositional*.[8]

PROPOSITION 49. *Let $\mathcal{C}$ be a set of propositional EPL clauses and φ a formula. Let $\mathcal{C}'$ be the set of clauses obtained from $\mathcal{C} \cup \{(\varphi\ (N\ 1))\}$. Then the valuation of the optimal refutation from $\mathcal{C}'$ is $Val(\varphi, \mathcal{F})$.*

Lastly, due to the impossibility to extend Propositions 1 and 2 (related to the principle of minimum specificity) to EPL, semantic evaluation cannot be extended either.

5.2 Full possibilistic logic

SPL and EPL knowledge bases are sets, i.e., conjunctions, of possibilistic formulas. We consider here another way to extend standard and extended possibilistic logic, namely by allowing disjunctions of possibilistic formulas, such as $(\varphi\ (\Pi\ \alpha)) \vee (\psi(N\ \beta))$.[9] For the sake of simplicity and brevity we will only consider here the propositional necessity-valued fragment of FPL, that we call NFPL.[10]

DEFINITION 50. The language of necessity-valued full possibilistic logic (NFPL) is generated recursively by the following rules:

- $(\varphi\ \alpha) \in$ NFPL, where φ is a propositional formula and $\alpha \in (0, 1]$.

[7] The knowledge base must be already in clausal form because putting a formula in EPL into clausal form does not preserve the inconsistency degree: for instance, if $\mathcal{F} = \{(a \vee b\ (N\ 0.7)), (\neg a \wedge \neg b\ (\Pi\ 0.4))\}$ and $\mathcal{C} = \{(a \vee b\ (N\ 0.7)), (\neg a\ (\Pi\ 0.4)), (\neg b\ (\Pi\ 0.4))\}$, then $Incons(\mathcal{C}) = (\Pi\ 0)$ whereas $Incons(\mathcal{F}) = (\Pi\ 0.4)$.

[8] Let $\mathcal{C} = \{(\forall x)p(x)\ (\Pi\ 0.8), (p(a)\ (N\ 1)), (p(b)\ (N\ 1))\}$. There is no $(\Pi\ \alpha)$-refutation from $\mathcal{C}$ – because $p(x)\ (\Pi\ 0.8)$ should be used twice in the refutation and $(\Pi\ 0.8) \odot (\Pi\ 0.8) = (\Pi\ 0))$ – whereas $Incons(\mathcal{C}) = (\Pi\ 0.8)$.

[9] Note that allowing for *negations* of possibilistic formulas does not bring much new, since $\neg(\varphi\ (N\ \alpha))$, expressing $not(N(\varphi) \geq \alpha)$ i.e. $N(\varphi) < \alpha$ or equivalently $\Pi(\neg\varphi) > 1 - \alpha$, which is equivalent to an EPL possibility-valued formula, *no matter if the inequality is strict*: indeed, the finiteness of the knowledge base implies that only finitely many different valuations are used, and thus $\Pi(\neg\varphi) > 1 - \alpha$ is equivalent to $\Pi(\neg\varphi) \geq \gamma$ for a suitable choice of γ (depending on $\mathcal{F}$). This is why, when defining full possibilistic logic, we only add to the language *disjunctions* of possibilistic formulas.

[10] This gives the following diagram ($\rightarrow$ meaning “more general than”):

$$\begin{tikzcd} & \text{FPL} \arrow[dl] \arrow[dr] & \\ \text{NFPL} \arrow[dr] & & \text{EPL} \arrow[dl] \\ & \text{SPL} & \end{tikzcd}$$

- $\forall \Phi, \Psi \in$ NFPL, $\Phi \bigwedge \Psi$ and $\Phi \bigvee \Psi$ are in NFPL.

DEFINITION 51 (semantics of NFPL).

- $\pi \models (\varphi\ \alpha)$ iff $N(\varphi) \geq \alpha$
- $\pi \models \Phi \bigwedge \Psi$ iff $\pi \models \Phi$ and $\pi \models \Psi$
- $\pi \models \Phi \bigvee \Psi$ iff $\pi \models \Phi$ or $\pi \models \Psi$
- $\Phi \models \Psi$ iff $\forall \pi$, $\pi \models \Phi$ implies $\pi \models \Psi$
- $Val(\Phi, \varphi) = Sup\{\alpha \mid \Phi \models (\varphi\ \alpha)\}$
- $Incons(\Phi) = Val(\Phi, \bot)$

The external disjunctive normal form (EDNF) of a NFPL formula Φ will be the equivalent formula having the form

$$\bigvee_i \bigwedge_j (\varphi_{i,j}\ \alpha_{i,j})$$

It is obtained as in classical propositional logic by distributing $\bigwedge$ on $\bigvee$. Note that the internal formulas $\varphi_{i,j}$ are not necessarily in normal form. We note $\Phi_i = \bigwedge_j (\varphi_{i,j}\ \alpha_{i,j})$; thus $\Phi \equiv \bigvee_i \Phi_i$, each Φ_i being a *standard possibilistic knowledge base*.

PROPOSITION 52. *Let Φ be a NFPL formula and $\Phi_1, \ldots \Phi_n$ the disjuncts of its external disjunctive normal form,* i.e., $\Phi \equiv \bigvee_i \Phi_i$. *Let π_{Φ_i} be the least specific possibility distribution satisfying Φ_i (after Proposition 1) and N_i the necessity measure induced by π_{Φ_i}. Then, $\Phi \models (\varphi\ \alpha)$ iff $\min_i N_i(\varphi) \geq \alpha$*

Proof. $\pi \models \Phi$ if and only if $\exists i$ such that $\pi \models \Phi_i$. Thus, $\Phi \models (\varphi\ \alpha)$ if and only if $\bigcup_i \{\pi \mid \pi \models \Phi_i\} \subseteq \{\pi \mid \pi \models (\varphi\ \alpha)$. Since $\pi \models \Phi_i$ iff $\pi \leq \pi_i$ by Proposition 1, we get the result. ■

As an immediate corollary, we get $Val(\varphi, \Phi) = \min_i Val(\varphi, \Phi_i)$ and $Incons(\Phi) = \min_i Incons(\Phi_i)$.

The problem is that in the worst case, putting Φ under external DNF leads to an exponential number of Φ_i's, hence the complexity gap when switching from SPL to NFPL. Interestingly, it lays in the second level of the polynomial hierarchy, like many problems in knowledge representation (including most nonmonotonic logics and belief revision – see for instance [Gottlob, 1992; Nebel, to appear; Cayrol and Lagasquie-Schiex, 1995]).

PROPOSITION 53. *Deduction in NFPL is in Π_2^P.*

Proof. let us call NFPL-DED the problem of deciding whether $(\psi\ \beta)$ is a consequence of a NFPL formula Φ. The membership of the complementary problem $\overline{\text{NFPL-CONS}}$ to Σ_2^P is given by the following nondeterministic algorithm:
1. guess Φ_i a conjunction of SPL formulas formed with a subset of the possibilistic formulas $\{(\varphi_k\ \alpha_k), k \in K\}$ appearing in Φ.
2. considering all $\{(\varphi_k\ \alpha_k), k \in K\}$ appearing in Φ and Φ_i as classical propositional variables p_k, show that $\Phi_i \models \Phi$;
3. show that $\Phi_i \not\models (\psi\ \beta)$.
Steps 2 and 3 require NP-oracles, hence $\overline{\text{NFPL-CONS}} \in \Sigma_2^P$, hence NFPL-CONS $\in \Pi_2^P$. ∎

6 CONCLUSION

6.1 *What else?*

There are several issues, related to the algorithmic issues of possibilistic logic, which have not been discussed in this chapter.

Fuzzy constraint satisfaction [Dubois *et al.*, 1996] computes preferred solutions with respect to a set of prioritized constraints in a way that is very similar to possibilistic semantic evaluation.

Possibilistic logic programming uses possibilistic logic as a programming language, which is particularly suited to dealing with uncertainty or with minmax optimization. Formal details about declarative and procedural semantics of possibilistic logic programs can be found in [Dubois *et al.*, 1991c] and extensions incorporating negation by failure can be found in the recent works of Wagner [Wagner, 1997; Wagner, to appear].

Drowning-free variants of possibilistic logic. A consequence of Proposition 5 is that in a SPL knowledge base $\mathcal{F}$, formulas whose valuation is not larger than $Incons(\mathcal{F})$ are completely useless for nontrivial deductions – this is the *drowning effect* [Benferhat *et al.*, 1993]. For instance, $\mathcal{F} = \{(a\ 1), (\neg a\ 0.8), (b\ 0.6)\}$ is equivalent to $\{(a\,1), (\perp\ 0.8)\}$, and thus $(b\,0.6)$ is drowned by the 0.8-inconsistency. In order to escape the drowning effect, the idempotent operation $\min$ in $\pi_{\mathcal{F}}(\omega) = \min_{\omega \models \neg\varphi_i} 1 - \alpha_i$ (Proposition 1) must be replaced by a non-idempotent operation. Two alternative operators, close to the qualitative spirit of $\min$ but not idempotent, have been proposed for defining drowning-free variants of possibilistic logic namely, leximin and discrimin [Dubois *et al.*, 1992b; Benferhat *et al.*, 1993]. As for complexity, recall (Proposition 13) that nonmonotonic possibilistic entailment is in $\Delta_2^P(\mathcal{O}(\log n))$; then the leximin variant is in $\Delta_2^P(\mathcal{O}(n))$ [Cayrol and Lagasquie-Schiex, 1995] and the discrimin variant is Π_2^P-complete [Nebel, 1991; Cayrol and Lagasquie-Schiex, 1995]. This means that escaping the drowning effect generates an increase of complexity; as to practical computation, drowning-free variants basically require computing some of the maximal consistent subsets of the knowledge base, which SPL does not.

There are also some generalizations of possibilistic logic that we did not discuss here, including possibilistic assumption-based truth maintenance systems, possibilistic logic with vague predicates [Dubois and Prade, 1987], lattice-based possibilistic logics where the set of certainty values is not any longer $[0, 1]$ but more generally a lattice [Dubois *et al.*, 1991b], possibility theory based on non-classical logics [Besnard and Lang, 1994; Boldrin and Sossai, 1995].

6.2 What for?

This chapter was written in a computational perspective, and therefore we said very little about the potential impact of the algorithmic issues we discussed. Among the applications of possibilistic logic we find:

1. nonmotonic reasoning [Dubois and Prade, 1991b; Dubois *et al.*, 1989], reasoning with default rules [Benferhat *et al.*, 1992];
2. belief revision [Dubois and Prade, 1992];
3. multisource reasoning [Dubois *et al.*, 1992a], inconsistency handling [Dubois *et al.*, 1992b], knowledge fusion [Benferhat *et al.*, 1997];
4. reasoning about action [Dubois *et al.*, 1994], planning under uncertainty [Costa Pereira *et al.*, 1997];
5. diagnosis [Cayrac *et al.*, 1995];
6. terminological logics [Hollunder, 1994];
7. temporal reasoning [Dubois *et al.*, 1991b], graded persistence [Driankov and Lang, 1993], timed stamped knowledge bases;
8. solving discrete optimization problems with minmax costs, reasoning with prioritized constraints [Lang, 1991b], qualitative decision making [Dubois *et al.*, 1997];
9. reasoning with graded obligations and permissions.

6.3 What next?

Algorithmic issues in possibilistic logic are related to their counterparts in related logics of formalisms for reasoning with uncertainty, some of which are considered in this book: probabilistic logics (Jaumard and Hansen, chapter 8), Dempster-Shafer logics (Kohlas and Haenni, chapter 6), default logics (Mengin, chapter 3).

It is possible to reformulate possibilistic logic into a much more general framework which encompasses other weighted logics of uncertainty (among which several variants of probabilistic logic and of belief function logics, as well as more "exotic" logics of uncertainty). This gives raise to so-called "weighted logics" [Dubois *et al.*, in preparation]. Intuitively, a weighted logic consists in the association of *valuations* from a given structure (generally a completely ordered lattice) to logical formulae; these valuations will be required to satisfy a list of given axioms depending on the particular weighted logic (for instance, axioms of possibility theory for possibilistic logic); furthermore, these valuations operate *upon* a given logical structure with which they are fully compatible (in this chapter, this logical structure was classical logic). The study of general automated deduction algorithms for specific classes of weighted logics is a promising topic. However, it appears that possibilistic logic enjoys many of its computational properties thanks to the idempotency of the $\min$ operator (see [Schiex *et al.*, 1995] for a similar general work in a constraint satisfaction framework).

Lastly, a point which was not discussed in this chapter was the design of *anytime* and *approximate* algorithms for possibilistic logic. As pointed out in [Williams, 1997] in a belief revision context, many of the algorithms given in this chapter can be used in an anytime way (which means that they provide an output which gets better and better with the actual execution time). An experimental study of their efficiencies remains to be done.

APPENDIX

We assume that complexity classes NP and coNP are known by the reader. We recall here three other classes, located between the two latter ones and PSPACE. The reader may consult Papadimitriou [Papadimitriou, 1994] for an extensive presentation. The polynomial hierarchy (containing Δ_2^P, Σ_2^P and Π_2^P) has been first defined by Stockmeyer [Stockmeyer, 1977]. These classes has proven to be of some great interest for Artificial Intelligence, and in particular, belief revision [Nebel, to appear; Eiter and Gottlob, 1992] and nonmonotonic reasoning [Cadoli, 1993; Cayrol and Lagasquie-Schiex, 1995; Gottlob, 1992].

- DP, or BH_2, is the set of all languages which are the intersection of a language in NP and a language in coNP. The canonical DP-complete problem is SAT-UNSAT: given two propositional formulas φ, ψ, decide that φ is satisfiable and ψ is not.

- $\Delta_2^P = \mathsf{P}^{\mathsf{NP}}$ [Stockmeyer, 1977] is the class of all decision problems that can be decided in polynomial time using NP-oracles; $\Delta_2^P(\mathcal{O}(logn))$ is the class of all problems which can be decided in polynomial time using a logarithmically many NP-oracles calls.

- $\Sigma_2^P = \mathsf{NP}^{\mathsf{NP}}$ [Stockmeyer, 1977] is the class of all decision problems that can be decided in polynomial time on a nondeterministic Turing machine using NP-oracles. The canonical Σ_2^P-complete problem is 2-QBF (where QBF stands for "Quantified Boolean Formula"): $\exists a_1 \ldots \exists a_m \forall b_1 \ldots \forall b_n F$, where the a_i's and b_j's are Boolean variables and F is a propositional formula built on them.

- $\Pi_2^P = \text{co-}\Sigma_2^P$. The canonical Π_2^P-complete problem is $2 - \overline{\text{QBF}}$: $\forall a_1 \ldots \forall a_m \exists b_1 \ldots \exists b_n F$.

ACKNOWLEDGEMENTS

Thanks to Simon Parsons and Mary-Anne Williams for helpful comments on a first draft of this paper.

IRIT, Université Paul Sabatier, Toulouse, France.

BIBLIOGRAPHY

[Benferhat, 1994] Salem Benferhat. *Raisonnement non-monotone et traitement de l'inconsistance en logique possibiliste*. *PhD Thesis*, Université Paul Sabatier, 1994.

[Benferhat *et al.*, 1993] Salem Benferhat, Claudette Cayrol, Didier Dubois, Jérôme Lang and Henri Prade. Inconsistency management and prioritized syntax-based entailment, *Proc. of the 13th International Joint Conference on Artificial Intelligence (IJCAI'93)*, 640-645.

[Benferhat *et al.*, 1992] Salem Benferhat, Didier Dubois and Henri Prade. Default rules and possibilistic logic. *Proceedings of the 3rd Conference on Knowledge Representation and Reasoning (KR'92)*, 673-684.

[Benferhat *et al.*, 1997] Salem Benferhat, Didier Dubois and Henri Prade. From semantic to syntactic approaches to information combination in possibilistic logic. In *Qualitative and Quantitative Practical Reasoning*, Lectures Notes in Artificial Intelligence 1244, Springer-Verlag, 1997. Long version to appear in *Aggregation of Evidence under Fuzziness* (B. Bouchon-Meunier, ed.), Physica Verlag.

[Besnard and Lang, 1994] Philippe Besnard and Jérôme Lang. Possibility and necessity functions over non-classical logics. *Proceedings of the 10th Int. Conf. on Uncertainty in Artificial Intelligence (UAI'94)*, Seattle, July 94 (Morgan Kaufman), 69-76.

[Boldrin and Sossai, 1995] Luca Boldrin and Claudio Sossai. An algebraic semantics for possibilistic logic. *Proceedings of the 11th Int. Conf. on Uncertainty in Artificial Intelligence (UAI'95)*, 27-35.

[Cadoli, 1993] Marco Cadoli and Marco Schaerf. A survey of complexity results for nonmonotonic logics. *Journal of Logic Programming* 17:127-160, 1993.

[Cayrac *et al.*, 1995] Didier Cayrac, Didier Dubois and Henri Prade. Practical model-based diagnosis with qualitative possibilistic uncertainty. *Proceedings of the 11th Int. Conf. on Uncertainty in Artificial Intelligence (UAI'95)*, 68-76.

[Cayrol and Lagasquie-Schiex, 1995] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Nonmonotonic syntax-based entailment: a classification of consequence relations. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty* (Proceedings of ECSQARU'95, Fribourg, July 95), Lectures Notes in Artificial Intelligence 946, Springer Verlag, 107-114.

[Chang and Lee, 1973] C.C. Chang, R.C.T. Lee. *Symbolic logic and mechanical theorem proving*, Academic Press, 1973.

[Davis and Putnam, 1960] M. Davis, H. Putnam. A computing procedure for quantification theory, *J. of the Assoc. for Computing Machinery* 7 (1960), 201-215.

[Costa Pereira *et al.*, 1997] C. da Costa Pereira, F. Garcia, J. Lang and R. Martin-Clouaire, "Possibilistic planning: representation and complexity", in *Recent Advances in Planning* (Sam Steel, Rachid Alami, eds.), Lectures Notes in Artificial Intelligence, Springer Verlag, 1997, 143-155. Long version in *International Journal of Intelligent Systems*, 12 (11/12), 1997, 935-962.

[Driankov and Lang, 1993] Dimiter Driankov and Jérôme Lang, Possibilistic decreasing persistence. *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence (UAI'93)*, Washington, Juillet 1993, 469-476.

[Dubois *et al.*, 1994] Didier Dubois, Florence Dupin de Saint-Cyr and Henri Prade. Updating, transition constraints and possibilistic Markov chains, *Lectures Notes in Computer Science* 945 (B. Bouchon-Meunier, R.R. Yager, L. Zadeh eds.), Springer-Verlag, 1994, 263-272.

[Dubois *et al.*, in preparation] Didier Dubois, Florence Dupin de Saint-Cyr, Jérôme Lang, Henri Prade and Thomas Schiex. Weighted logics of uncertainty. In preparation.

[Dubois *et al.*, 1996] Didier Dubois, Hélène Fargier and Henri Prade. Possibility theory in constraint satisfaction problems: handling priority, preference and uncertainty, *Applied Intelligence* 6, 287-309, 1996.

[Dubois *et al.*, 1987] Didier Dubois, Jérôme Lang and Henri Prade. Theorem proving under uncertainty: a possibility theory-based approach. *Proc. of the 10th International Joint Conference on Artificial Intelligence (IJCAI'87)*, 484-486.

[Dubois *et al.*, 1989] Didier Dubois, Jérôme Lang and Henri Prade. Automated reasoning using possibilistic logic: semantics, belief revision, and variable certainty weights. *IEEE Trans. on Data and Knowledge Engineering*, 1994.

[Dubois *et al.*, 1991a] Didier Dubois, Jérôme Lang and Henri Prade. Fuzzy sets in approximate reasoning – Part II: logical approaches. *Fuzzy Sets and Systems*, 40:203-244, 1991.

[Dubois *et al.*, 1991b] Didier Dubois, Jérôme Lang and Henri Prade. Timed possibilistic logic. *Fundamenta Informaticae*, XV:211-234, 1991.

[Dubois *et al.*, 1991c] Didier Dubois, Jérôme Lang and Henri Prade. Towards possibilistic logic programming. *Proc. Inter. Conf. on Logic Programming* (ICLP'91), Paris, June 1991, MIT Press, 581-595.

[Dubois *et al.*, 1992a] Didier Dubois, Jérôme Lang and Henri Prade. Dealing with multi-source information in possibilistic logic. *Proc. of the 10th European Conf. on Artificial Intelligence (ECAI'92)*, Vienna, Aug. 1992, 38-42.

[Dubois *et al.*, 1992b] Didier Dubois, Jérôme Lang and Henri Prade. Inconsistency in possibilistic knowledge bases: to live or not live with. In *Fuzzy Logic for the Management of Uncertainty* (L. Zadeh, J. Kacprzyk, eds.), Wiley, 1992, 335-352.

[Dubois *et al.*, 1994] Didier Dubois, Jérôme Lang and Henri Prade. Possibilistic logic. *Handbook of Logic in Artificial Intelligence and Logic Programming* (D.M. Gabbay, C.J. Hogger, J.A. Robinson, eds.), Vol. 3, 439-513, Oxford University Press.

[Dubois and Prade, 1987] Didier Dubois and Henri Prade. Necessity measures and the resolution principle, *IEEE Trans. on Systems, Man and Cybernetics*, 17:474-478, 1987.

[Dubois and Prade, 1988] Didier Dubois and Henri Prade. An introduction to possibilistic and fuzzy logics (with discussions and reply). In *Non-standard Logics for Automated Reasoning* (P. Smets, A. Mamdani, D. Dubois, H. Prade, eds.), Academic Press, 287-315 & 321-326.

[Dubois and Prade, 1990] Didier Dubois and Henri Prade. Resolution principles in possibilistic logic. *Int. Journ. of Approximate Reasoning*, 4(1):1-21, 1990.

[Dubois and Prade, 1991a] Didier Dubois and Henri Prade. Epistemic entrenchment and possibilistic logic. *Artificial Intelligence*, 50:223-239, 1991.

[Dubois and Prade, 1991b] Didier Dubois and Henri Prade. Possibilistic logic, preferential models, nonmonotonicity and related issues. *Proc. of the 12th International Joint Conference on Artificial Intelligence (IJCAI'91)*, 419-424.

[Dubois and Prade, 1992] Didier Dubois and Henri Prade. Belief change and possibility theory. In P. Gärdenfors, ed., *Belief Revision*, 142-182, Cambridge University Press, 1992.

[Dubois *et al.*, 1997] D. Dubois, H. Prade and R. Sabbadin. A possibilistic logic machinery for qualitative decision. In *Proc. of the AAAI 1997 Spring Symposium Series (Qualitative Preferences in Deliberation and Practical Reasoning)*, Standford University, California, March 24-26, 1997.

[Eiter and Gottlob, 1992] Thomas Eiter and Georg Gottlob. On the complexity of propositional knowledge base revisions, updates, and counterfactuals. *Artificial Intelligence* 57:227-270, 1992.

[Farreny, 1997] Henri Farreny. *Recherche heuristiquement ordonnée dans les graphes d'états*. Masson, 1995. English version in Technical Report IRIT, Université Paul Sabatier, Toulouse, 1997, [Fariñas del Cerro and Herzig, 1991] Luis Fariñas del Cerro and Andreas Herzig. A modal analysis of possibility theory. *Fundamentals of Artificial Intelligence Research*, 11-18, Lecture Notes in Computer Science, Vol. 535, 1991.

[Fariñas del Cerro *et al.*, 1994] Luis Fariñas del Cerro, Andreas Herzig and Jérôme Lang. From ordering-based nonmonotonic reasoning to conditional logics. *Artificial Intelligence* 66 (1994), 375-393.

[Froidevaux and Grossetête, 1990] Christine Froidevaux and Christine Grossetête. Graded default theories for uncertainty. *Proc. of the 10th European Conf. on Artificial Intelligence (ECAI'90)*, Stockholm, Aug. 1990, 283-288.

[Froidevaux and Mengin,] Christine Froidevaux and Jérôme Mengin. A theorem prover for free graded default theories. Tech. Report.

[Gabbay, 1991] Dov Gabbay. *Labelled Deductive Systems*, Oxford University Press, 1991.

[Gottlob, 1992] Georg Gottlob. Complexity results for nonmonotonic logics. *Journal for Logic and Computation*, 2(3):397-425, 1992.

[Jeannicot *et al.*, 1988] Serge Jeannicot, Laurent Oxusoff and Antoine Rauzy. Evaluation sémantique: une propriété de coupure pour rendre efficace la procédure de Davis et Putnam, *Revue d'Intelligence Artificielle*, 2 (1):41-60, 1988.

[Hollunder, 1994] Bernhard Hollunder. An alternative proof method for possibilistic logic and its application to terminological logics. *Proceedings of the 10th Int. Conf. on Uncertainty in Artificial Intelligence (UAI'94)*, Seattle, July 94 (Morgan Kaufman), 327-335.

[Knuth and Moore, 1975] D.E. Knuth and R.W Moore, An analysis of alpha-beta pruning, *Artificial Intelligence* 6 (1975), 203-326.

[Lang, 1990] Jérôme Lang. Semantic evaluation in possibilistic logic. In *Uncertainty in Knowledge-Based Systems* (B. Bouchon, R. Yager, L. Zadeh, eds.), Lecture Notes in Computer Science, Vol. 521, Springer Verlag, 1991, 260-268.

[Lang, 1991a] Jérôme Lang. *Logique possibiliste: aspects formels, déduction automatique et applications*, *PhD Thesis*, Université Paul Sabatier, 1991.

[Lang, 1991b] Jérôme Lang. Possibilistic logic as a framework for min-max discrete optimisation problems and prioritized constraints, *Fundamentals of Artificial Intelligence Research*, 112-126, Lecture Notes in Computer Science, Vol. 535, 1991.

[Lang, 1991c] Jérôme Lang, Didier Dubois and Henri Prade. A logic of graded possibility and certainty coping with partial inconsistency. *Proc. of the 7th Inter. Conf on Uncertainty in Artificial Intelligence (UAI'91)*, Morgan Kaufmann, 188-196.

[Lee, 1972] R.C.T Lee. Fuzzy logic and the resolution principle, *Journ. of the ACM*, 19:109-119, 1972.

[Liau and Lin, 1996] Churn-Jung Liau and Bertrand I-Peng Lin. Possibilistic reasoning: a mini-survey and uniform semantics. *Artificial Intelligence* 88 (1996), 163-193.

[Loveland, 1978] Donald W. Loveland. *Automated theorem proving: a logical basis*, North-Holland, 1978.

[Nebel, 1991] Bernhard Nebel. Belief revision and default reasoning: syntax-based approaches. *Proceedings of the 2nd Conference on Knowledge Representation and Reasoning (KR'91)*, 417-428.

[Nebel, to appear] Bernhard Nebel. How hard is it to revise a belief base? In *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, Volume 3 - Belief Change, pp. 77–146, Kluwer, 1999.

[Nilsson, 1986] Nils J. Nilsson. Probabilistic logic, *Artificial Intelligence*, 28:71-87, 1986.

[Papadimitriou, 1994] Christos H. Papadimitriou. *Computational complexity*, Addison-Wesley, 1994.

[Saffiotti, 1992] Alessandro Saffiotti. A belief function logic, *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI"92)*, 642-647.

[Schiex *et al.*, 1995] Thomas Schiex, Hélène Fargier and Gérard Verfaillie. Valued constraint satisfaction problems, *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, 631-637.

[Stockmeyer, 1977] Larry Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science* 3, 1-22, 1977.

[Wagner, 1997] Gerd Wagner. A Logical Reconstruction of Fuzzy Inference in Databases and Logic Programs, *Proceedings of IFSA'97*.

[Wagner, to appear] Gerd Wagner. Negation in Fuzzy and Possibilistic Logic Programs. To appear.

[Williams, 1997] Mary-Anne Williams. Anytime belief revision. *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI'97)*, 74-79.

[Yager, 1986] Ronald R. Yager. Paths of least resistance in possibilistic production systems. *Fuzzy Sets and Systems* 10:121-132, 1986.

[Zadeh, 1965] Lotfi A. Zadeh. Fuzzy Sets, *Information and Control*, 8:338-353, 1965.

[Zadeh, 1978] Lotfi A. Zadeh. Fuzzy sets as a basis for theory of possibility, *Fuzzy Sets and Systems* 1(1):3-28, 1978.