Possibilistic Stable Models

Pascal Nicolas and **Laurent Garcia** and **Igor Stéphan**
LERIA, University of Angers, France
{pascal.nicolas, laurent.garcia, igor.stephan}@univ-angers.fr

Abstract

In this work, we define a new framework in order to improve the knowledge representation power of Answer Set Programming paradigm. Our proposal is to use notions from possibility theory to extend the stable model semantics by taking into account a certainty level, expressed in terms of necessity measure, on each rule of a normal logic program. First of all, we introduce possibilistic definite logic programs and show how to compute the conclusions of such programs both in syntactic and semantic ways. The syntactic handling is done by help of a fix-point operator, the semantic part relies on a possibility distribution on all sets of atoms and we show that the two approaches are equivalent. In a second part, we define what is a possibilistic stable model for a normal logic program, with default negation. Again, we define a possibility distribution allowing to determine the stable models.

1 Introduction

Answer set programming (ASP) is an appropriate formalism to represent various problems issued from Artificial Intelligence and arising when available information is incomplete as in non-monotonic reasoning, planning, diagnosis... From a global view, ASP is a general paradigm covering different declarative semantics for different kinds of logic programs. One is the *stable model semantics* [Gelfond and Lifschitz, 1988] for definite logic programs augmented with default negation. In ASP, information is encoded by logical rules and solutions are obtained as sets of models. Each model is a minimal set of atoms representing sure informations (some facts) and deductions obtained by applying by default some rules. So, conclusions rely on present and absent information, they form a coherent set of hypotheses and represent a rational view on the world described by the rules. Thus, in whole generality there is not a unique set of conclusions but maybe many ones and each conclusion is no longer absolutely sure but only plausible and more or less certain.

Possibilistic logic [Dubois *et al.*, 1995] is issued from Zadeh's possibility theory [Zadeh, 1978], which offers a framework for representation of states of partial ignorance owing to the use of a dual pair of possibility and necessity measures. Possibility theory may be quantitative or qualitative [Dubois and Prade, 1998] according to the range of these measures which may be the real interval $[0, 1]$, or a finite linearly ordered scale as well. Possibilistic logic provides a sound and complete machinery for handling qualitative uncertainty with respect to a semantics expressed by means of possibility distributions which rank-order the possible interpretations. Let us mention that in possibilistic logic we deal with uncertainty by means of classical 2-valued interpretations that can be more or less certain.

The aim of our work is to introduce the concepts of possibilistic logic into ASP in order to deal with uncertainty. This can be illustrated by the following example of a medical treatment in which a patient is suffering from two diseases. Each disease can be cured by one drug but the two drugs are incompatible. The program $P = \{dr1 \leftarrow di1, not\ dr2.$, $dr2 \leftarrow di2, not\ dr1.$, $c1 \leftarrow dr1, di1.$, $c2 \leftarrow dr2, di2.$, $di1.$, $di2.\}$ means that the drug $dr1$ (resp. $dr2$) is given to a patient if he suffers from the disease $di1$ (resp. $di2$) except if he takes the drug $dr2$ (resp. $dr1$); if a patient suffering from disease $di1$ (resp. $di2$) takes the drug $dr1$ (resp. $dr2$) then he is cured $c1$ (resp. $c2$) of his disease; the patient suffers from the diseases $di1$ and $di2$. From this program, two stable models $\{di1, di2, dr1, c1\}$ and $\{di1, di2, dr2, c2\}$ are obtained. However, it seems interesting for a doctor to evaluate the certainty of the efficiency of the medical treatment. So, it seems natural to combine the certainty degrees of rules allowing to determine the level of certainty of the conclusions.

This section continues with a theoretical background on possibilistic logic and ASP. Section 2 introduces possibilistic definite logic programs and their possibilistic models. Section 3 extends our approach to possibilistic stable models for logic programs with default negation. Last, section 4 links our work to some others in the same area and gives some perspectives.

Possibilistic logic. Possibilistic logic, in the necessity-valued case, handles pairs of the form (p, α) where p is a classical logic formula and α is an element of a totally ordered set. The pair (p, α) expresses that the formula p is certain at least to the level α. A possibilistic base is denoted $\Sigma = \{(p_i, \alpha_i)\}_{i \in I}$. Formulas with degree 0 are not explicitly represented in the knowledge base since only beliefs which are somewhat accepted are useful. The higher is the weight,

Dagstuhl Seminar Proceedings 05171
Nonmonotonic Reasoning, Answer Set Programming and Constraints
http://drops.dagstuhl.de/opus/volltexte/2005/264

the more certain is the formula. This degree α is evaluated by a necessity measure and it is not a probability. Thus, numerical values are not an absolute evaluation (like it is in probability theory) but induce a certainty (or confidence) scale. Moreover, let us note that these values are determined by the expert providing the knowledge base or they are automatically given if, as we can imagine, the rules and their confidence degrees result from a knowledge discovery process.

The basic element of possibility theory is the possibility distribution π which is a mapping from Ω, the interpretation set, to the interval $[0, 1]$. $\pi(\omega)$ represents the degree of compatibility of the interpretation ω with the available information (or beliefs) about the real world. By convention, $\pi(\omega) = 0$ means that ω is impossible, and $\pi(\omega) = 1$ means that nothing prevents ω from being the real world (ω is consistent with all the available beliefs, it is a model of Σ). When $\pi(\omega) > \pi(\omega')$, ω is a preferred candidate to ω' for being the real state of the world. From a possibility distribution π, we can define two different ways of rank-ordering formulas of the language. This is obtained using two mappings grading respectively the possibility and the certainty of a formula p: $\Pi(p) = max\{\pi(\omega) \mid \omega \models p\}$ is the possibility (or consistency) degree which evaluates the extent to which p is consistent with the available beliefs expressed by π [Zadeh, 1978]; $N(p) = 1 - \Pi(\neg p) = 1 - max\{\pi(\omega) \mid \omega \not\models p\}$ is the necessity (or certainty) degree which evaluates the extent to which p is entailed by the available beliefs.

Moreover, given a base Σ, only the possibility distributions respecting $\forall(p, \alpha) \in \Sigma,\ N(p) \geq \alpha$ are significant and called *compatible*. A possibility distribution π is said to be the least specific one between all compatible distributions if there is no possibility distribution π', with $\pi' \neq \pi$, compatible with Σ such that $\forall\omega, \pi'(\omega) \geq \pi(\omega)$. The least specific possibility distribution π_Σ always exists [Dubois *et al.*, 1995] and is defined by:

if ω is a model of Σ, then $\pi_\Sigma(\omega) = 1$
else $\pi_\Sigma(\omega) = 1 - \max\{\alpha \mid \omega \not\models p, (p, \alpha) \in \Sigma\}$.

Stable Model Semantics. ASP is concerned by different kinds of logic programs and different semantics. In our work we deal with *normal logic programs*, interpreted by *stable model semantics* [Gelfond and Lifschitz, 1988]. We consider given a non empty set of atoms $\mathcal{X}$ that determines the language of the programs[1]. A *normal logic program* is a set of rules of the form: $c \leftarrow a_1,\ \ldots,\ a_n,\ not\ b_1,\ \ldots,\ not\ b_m$. where $n \geq 0$, $m \geq 0$, $\{a_1, \ldots, a_n, b_1, \ldots, b_m, c\} \subseteq \mathcal{X}$. A term like $not\ b$ is called a *default negation*. The intuitive meaning of such a rule is: "if you have all the a_i's and no b_j's then you can conclude c". For such a rule r we use the following notations (extended to a rule set as usual): the positive prerequisites of r, $body^+(r) = \{a_1,\ \ldots,\ a_n\}$; the negative prerequisites of r, $body^-(r) = \{b_1,\ \ldots,\ b_m\}$; the conclusion of r, $head(r) = c$ and the positive projection of r, $r^+ = head(r) \leftarrow body^+(r)$. If a program P does not contain any default negation (ie: $body^-(P) = \emptyset$), then P is a *definite logic program* and it has one minimal Herbrand model $Cn(P)$. The *reduct* P^X of a program P wrt. an atom set X is the definite logic program defined by : $P^X = \{r^+ \mid r \in P,\ body^-(r) \cap X = \emptyset\}$ and it is the core of the definition of a *stable model*.

Definition 1.1 *[Gelfond and Lifschitz, 1988] Let P be a normal logic program and S an atom set. S is a stable model of P if and only if $S = Cn(P^S)$.*

Note that a program may have one or many stable models or not at all. In this last case we say that the program is *inconsistent*, otherwise it is *consistent*. When an atom belongs at least to one stable model of P it is called a *credulous* consequence of P and when it belongs to every stable model of P, it is called a *skeptical* consequence of P.

Let A be an atom set, r be a rule and P be a program (definite or normal). We say that r is *applicable* in A if $body^+(r) \subseteq A$ and we denote $App(P, A)$ the subset of P of its applicable rules in A. A satisfies r (or r is satisfied by A), denoted $A \models r$, if r is applicable in $A \Rightarrow head(r) \in A$. A is *closed* under P if $\forall r \in P, A \models r$. P is *grounded* if it can be ordered as the sequence $\langle r_1, \ldots, r_n\rangle$ such that $\forall i, 1 \leq i \leq n, r_i \in App(P, head(\{r_1, \ldots, r_{i-1}\}))$. $Cn(P)$, the least (Herbrand) model of a definite logic program P, is the smallest atom set closed under P and it can be computed as the least fix-point of the following consequence operator $T_P : 2^{\mathcal{X}} \rightarrow 2^{\mathcal{X}}$ such that: $T_P(A) = head(App(P, A))$.

By this way, we can establish the next result that clarifies the links between the least model A of a program P and the rules producing it. We see that A is underpinned by a set of applicable rules, $App(P, A)$, that satisfies a stability condition and that is grounded. These two features will be used in the sequel to define the core of our work: a possibility distribution over atom sets induced by a definite logic program.

Lemma 1.1 *Let P be a definite logic program and A be an atom set,*

$$\text{A is the least Herbrand model of P} \iff \begin{cases} A = head(App(P, A)) \\ App(P, A) \text{ is grounded} \end{cases}$$

2 Possibilistic Definite Logic Programs

We consider given a finite set of atoms $\mathcal{X}$ and a finite, totally ordered set of necessity values $\mathcal{N} \subseteq]0, 1]$. Then, a *possibilistic atom* is a pair $p = (x, \alpha) \in \mathcal{X} \times \mathcal{N}$. We denote by $p^* = x$ the classical projection of p and by $n(p) = \alpha$ its necessity degree. These notations can also be extended to a *possibilistic atom set* (p.a.s.) A that is a set of possibilistic atoms where every atom x occurs at most one time in A.

A *possibilistic definite logic program* (p.d.l.p.) is a set of *possibilistic rules* of the form: $r = (c \leftarrow a_1,\ \ldots,\ a_n.\ , \alpha)$ $n \geq 0, \{a_1,\ \ldots,\ a_n,\ c\} \subseteq \mathcal{X}, \alpha \in \mathcal{N}$. The *classical projection* of the possibilistic rule is $r^* = c \leftarrow a_1,\ \ldots,\ a_n$. $n(r) = \alpha$ is a necessity degree representing the certainty level of the information described by r. If R is a set of possibilistic rules, then $R^* = \{r^* \mid r \in R\}$ is the definite logic program obtained from P by forgetting all the necessity values.

Let us recall that a possibilistic logic base is a compact representation of the possibility distribution defined on interpretations representing the information. Indeed, the treatment of

[1] In the sequel, if $\mathcal{X}$ is not explicitly given, it is supposed to be the set of all atoms occurring in the considered program.

the base in a syntactic way (in terms of formulas and necessity degrees) leads to the same results as the treatment done in a semantic way (in terms of interpretations and possibility distribution). In our framework, the same situation occurs as it is shown in the next two subsections. Firstly, we define a semantic handling of a program that is defined in term of a possibility distribution over all atom sets. Secondly, we provide a syntactic deduction process that leads to the same results as the ones given by the possibility distribution.

2.1 Model Theory for Possibilistic Definite Logic Programs

From a possibilistic definite logic program P, we can determine, as it is done in possibilistic logic, some possibility distributions defined on all the sets in $2^{\mathcal{X}}$ and that are compatible with P. Let us note that there is a correspondence between interpretations in propositional logic and atom sets in ASP. Like in possibilistic logic, the possibility degree of an atom set is determined by the necessity degrees of the rules of the program that are not satisfied by this set. The satisfiability of a rule r is based on its applicability wrt. an atom set A, so $A \not\models r$ iff $body^+(r) \subseteq A$ and $head(r) \notin A$ (see section 1). But, we have to notice that the contradiction of a rule is not enough to determine the possibility degree of a set since, in ASP, it is important to take into account the groundedness and stability notions (see lemma 1.1). Firstly, the set $A = \{a, b\}$ satisfies every rule in $P = \{a \leftarrow b.,\ b \leftarrow a.\}$, but it is not the least model of P because the groundedness is not satisfied. Secondly, the set $A' = \{a, b, d\}$ satisfies every rule in $P' = \{a., b \leftarrow a., d \leftarrow c.\}$ but it is not the least model of P' because d can not be produced by any rule from P' applicable in A'. In these two cases, the possibility of A and A' must be 0 since they can not be the least model at all, even if they satisfy every rule in their respective associated programs.

Definition 2.1 *Let P be a p.d.l.p. and $\pi : 2^{\mathcal{X}} \rightarrow [0,1]$ be a possibility distribution. π is compatible with P if*

$$\begin{cases} A \not\subseteq head(App(P^*, A)) \Rightarrow \pi(A) = 0 \\ App(P^*, A) \text{ is not grounded } \Rightarrow \pi(A) = 0 \\ A \text{ is the least model of } P^* \Rightarrow \pi(A) = 1 \\ \text{otherwise } \pi(A) \leq 1 - \max_{r \in P}\{n(r) \mid A \not\models r^*\} \end{cases}$$

The necessity degree attached to each rule defines only a lower bound (and not an exact value) of the certainty of the rule. So, as recalled in section 1, many possibility distributions are compatible with these degrees. But, we are only interested by the least informative one, that is the *least specific* one, whose characterization is given below.

Proposition 2.1 *Let P be a p.d.l.p. then $\pi_P : 2^{\mathcal{X}} \rightarrow [0,1]$ defined by*

1. *if $A \not\subseteq head(App(P^*, A))$ then $\pi_P(A) = 0$*
2. *if $App(P^*, A)$ is not grounded then $\pi_P(A) = 0$*
3. *otherwise*
   - *if $\forall r \in P, A \models r^*$ then $\pi_P(A) = 1$*
   - *otherwise $0 \leq \pi_P(A) < 1$ and* $\pi_P(A) = 1 - \max_{r \in P}\{n(r) \mid A \not\models r^*\}$

is the least specific possibility distribution.

The definition of π_P ensures that it is compatible with P. The third item ranks the sets which may be solutions with respect to the weights of the falsified rules.

Proposition 2.2 *Let P be a p.d.l.p. and $A \subseteq \mathcal{X}$ be an atom set, then*

1. $\pi_P(A) = 1 \iff A = Cn(P^*)$
2. $A \supset Cn(P^*) \Rightarrow \pi_P(A) = 0$
3. $Cn(P^*) \neq \emptyset \Rightarrow \pi_P(\emptyset) = 1 - \max_{r \in P}\{n(r) \mid \emptyset \not\models r^*\}$

Now, we can give the definition of inference that is, in the framework of ASP, the evaluation of the necessity degree of each atom of the universe.

Definition 2.2 *Let P be a p.d.l.p. and π_P the least specific possibility distribution compatible with P, we define the two dual possibility and necessity measures such that:*

$$\Pi_P(x) = \max_{A \in 2^{\mathcal{X}}}\{\pi_P(A) \mid x \in A\}$$
$$N_P(x) = 1 - \max_{A \in 2^{\mathcal{X}}}\{\pi_P(A) \mid x \notin A\}$$

$\Pi_P(x)$ gives the level of consistency of x with respect to the p.d.l.p. P and $N_P(x)$ evaluates the level at which x is inferred from the p.d.l.p. P. This is closely related to the definitions of possibilistic logic. For instance, whenever an atom x belongs to the least model of the classical program its possibility is total.

Proposition 2.3 *Let P be a p.d.l.p., $Cn(P^*)$ the least model of P^* and $x \in \mathcal{X}$ then :*

1. $x \in Cn(P^*) \Rightarrow \Pi_P(x) = 1$ *and* $x \notin Cn(P^*) \Rightarrow \Pi_P(x) = 0$
2. $x \notin Cn(P^*) \iff N_P(x) = 0$
3. $x \in Cn(P^*) \Rightarrow$ $N_P(x) = \min_{A \subset Cn(P^*)}\{\max_{r \in P}\{n(r) \mid A \not\models r^*\} \mid x \notin A\}$
4. *Let P' be a p.d.l.p., $P \subseteq P' \Rightarrow N_P(x) \leq N_{P'}(x)$.*

Furthermore, the necessity measure allows us to introduce the following definition of a *possibilistic model* of a p.d.l.p.

Definition 2.3 *Let P be a possibilistic definite logic program, then the set*

$$\Pi\mathcal{M}(P) = \{(x, N_P(x)) \mid x \in \mathcal{X}, N_P(x) > 0\}$$

is its possibilistic model.

Proposition 2.4 *Let P be a p.d.l.p. then : $(\Pi\mathcal{M}(P))^*$ is the least model of P^*.*

Example 2.1 *Let us take the following program $P = \{(a., 0.8),\ (b \leftarrow a., 0.6),\ (d \leftarrow a., 0.5),\ (d \leftarrow c., 0.9)\}$. The least specific possibility distribution induced by P is null for every atom set included in $\{a, b, c, d\}$ except for $\pi_P(\emptyset)$ = 0.2, $\pi_P(\{a\})$ = 0.4, $\pi_P(\{a, b\})$=0.5, $\pi_P(\{a, d\})$=0.4 and $\pi_P(\{a, b, d\})$=1 (the least model of P^*). So, we can compute the possibility of each atom: $\Pi_P(a) = \Pi_P(b) = \Pi_P(d) = 1$ and $\Pi_P(c) = 0$ and its certainty in term of necessity degree: $N_P(a)$ = 0.8, $N_P(b)$ = 0.6, $N_P(c)$ = 0, $N_P(d)$ = 0.5. Thus, $\Pi\mathcal{M}(P) = \{(a, 0.8), (b, 0.6), (d, 0.5)\}$ is the possibilistic model of P.*

2.2 Fix-point Theory for Possibilistic Definite Logic Programs

Definitions exposed in this subsection are closely related to what can be found in [Dubois *et al.*, 1991]. But here, we adopt an ASP point of view and thus we use atom sets instead of classical interpretations since the underlying possibility distribution is defined on atom sets.

Definition 2.4 *Consider $\mathcal{A} = 2^{\mathcal{X}\times\mathcal{N}}$ the finite set of all p.a.s. induced by $\mathcal{X}$ and $\mathcal{N}$. $\forall A, B \in \mathcal{A}$, we define:*

$$\begin{array}{lll} A \sqcap B & = \{(x, \min\{\alpha,\beta\}), (x,\alpha) \in A, (x,\beta) \in B\} \\ A \sqcup B & = \{(x,\alpha) \mid (x,\alpha) \in A, x \notin B^*\} \\ & \cup\{(x,\beta) \mid x \notin A^*, (x,\beta) \in B\} \\ & \cup\{(x, \max\{\alpha,\beta\}) \mid (x,\alpha) \in A, (x,\beta) \in B\} \\ A \sqsubseteq B & \iff \left\{ \begin{array}{l} A^* \subseteq B^*, \text{ and } \forall a, \alpha, \beta, \\ (a,\alpha) \in A \wedge (a,\beta) \in B \Rightarrow \alpha \le \beta \end{array} \right. \end{array}$$

Proposition 2.5 *$\langle \mathcal{A}, \sqsubseteq \rangle$ is a complete lattice.*

Definition 2.5 *Let $r = (c \leftarrow a_1,\ \ldots,\ a_n., \alpha)$ be a possibilistic rule and A be a p.a.s.,*

- *r is α-applicable in A if $body(r^*) = \emptyset$*
- *r is β-applicable in A if $\beta = \min\{\alpha, \alpha_1, \ldots, \alpha_n\}$, $\{(a_1,\alpha_1), \ldots, (a_n,\alpha_n)\} \subseteq A$,* [2]
- *r is 0-applicable otherwise.*

And, for a given p.d.l.p. P and an atom x, $App(P, A, x) = \{r \in P \mid head(r^*) = x, r \text{ is } \nu\text{-applicable in } A, \nu > 0\}$

The applicability degree of a rule r captures the certainty of the inference process realized when r is applied wrt. a p.a.s. A. If the body of r is empty, then r is applicable with its own certainty. If the body of r is not verified (not satisfied by A), then r is not at all applicable. Otherwise, the applicability level of r depends on the certainty level of the propositions inducing its groundedness and on its own necessity degree. Firstly, the necessity degree of a conjunction (the rule body) is the minimal value of the necessity values of subformulae (atoms) involved in it. Secondly, the certainty of a rule application is the minimal value between the rule certainty and the certainty of the rule body. This approach is similar to the resolution principle in possibilistic logic [Dubois *et al.*, 1995].

Definition 2.6 *Let P be a p.d.l.p. and A be a possibilistic atom set. The* immediate possibilistic consequence operator *ΠT_P maps a p.a.s. to another one by this way:* $\Pi T_P(A) =$

$$\left\{ \begin{array}{l} (x,\delta) \mid x \in head(P^*), App(P, A, x) \neq \emptyset, \\ \delta = \max_{r \in App(P,A,x)} \{\nu \mid r \text{ is } \nu\text{-applicable in } A\} \end{array} \right\}$$

then the iterated operator ΠT_P^k is defined by

$$\Pi T_P^0 = \emptyset \quad \text{and} \quad \Pi T_P^{n+1} = \Pi T_P(\Pi T_P^n), \forall n \ge 0 \quad .$$

Here, we can remark that if one conclusion is obtained by different rules, its certainty is equal to the greatest certainty with which it is obtained in each case (operator max). Again, it is in accordance with possibilistic resolution principle.

Proposition 2.6 *Let P be a p.d.l.p., then ΠT_P has a least fix-point $\sqcup_{n\ge 0}\Pi T_P^n(\emptyset)$ that we called the set of* possibilistic consequences *of P and we denote it by $\Pi Cn(P)$.*

[2] For two p.a.s. A and B, $A \subseteq B$ means the classical set inclusion and has not to be confused with the order $\sqsubseteq$ of def. 2.4.

Example 2.2 *Let P be the p.d.l.p. of example 2.1, then $\Pi T_P^0(\emptyset) = \emptyset$, $\Pi T_P^1(\emptyset) = \{(a, 0.8)\}$, $\Pi T_P^2(\emptyset) = \{(a, 0.8), (b, 0.6), (d, 0.5)\}$, $\Pi T_P^3(\emptyset) = \{(a, 0.8), (b, 0.6), (d, 0.5)\} = \Pi T_P^k(\emptyset), \forall k > 3$. So, $\Pi Cn(P) = \{(a, 0.8), (b, 0.6), (d, 0.5)\}$.*

Theorem 2.1 *Let P be a p.d.l.p., then $\Pi Cn(P) = \Pi\mathcal{M}(P)$.*

As illustrated and formalized above, our operator ΠT_P can be used to compute the possibilistic model of a p.d.l.p. This result shows the equivalence between the syntactic and semantic approaches in our framework, whereas in [Dubois *et al.*, 1991] only a syntactic treatment of a p.d.l.p. is proposed.

3 Possibilistic Normal Logic Programs

3.1 Possibilistic Normal Logic Programs and Possibilistic Stable Models

Here, we extend our proposal to non-monotonic reasoning by allowing default negation in programs and we formalize the notion of *possibilistic stable model*. It extends the stable model semantics by taking into account the necessity degree in the rules of a given *possibilistic normal logic program* (p.n.l.p.). Such a program is a finite set of rules of the form : $(c \leftarrow a_1,\ \ldots,\ a_n,\ not\ b_1,\ \ldots,\ not\ b_m.,\ \alpha)\ n \ge 0, m \ge 0$ for which we just have to precise that $\forall i, b_i \in \mathcal{X}$, all the rest being the same as for a p.d.l.p. (see the begining of section 2).

As in the classical case without necessity value, we need to define what is the reduction of a program before introducing the *possibilistic stable model semantics*.

Definition 3.1 *Let P be a p.n.l.p. and A be an atom set. The* possibilistic reduct *of P wrt. A is the p.d.l.p. $P^A = \{((r^*)^+, n(r)) \mid r \in P, body^-(r) \cap A = \emptyset\}$.*

Definition 3.2 *Let P be a p.n.l.p. and S a p.a.s. S is a possibilistic stable model of P if and only if $S = \Pi Cn(P^{(S^*)})$.*

By analogy with classical normal logic programs (without necessity values attached to rules) we say that a p.n.l.p. P is *consistent* if P has at least one possibilistic stable model. Otherwise P is said to be *inconsistent*. Furthermore when a possibilistic atom belongs at least to one possibilistic stable model of P it is called a *credulous* possibilistic consequence of P and when it belongs to every possibilistic stable model of P, it is called a *skeptical* possibilistic consequence of P.

Example 3.1 *We are now able to represent the example of the introductory section with some levels of certainty by $P = \{(dr1 \leftarrow di1, not\ dr2., 1), (dr2 \leftarrow di2, not\ dr1., 1), (c1 \leftarrow dr1, di1., 0.7), (c2 \leftarrow dr2, di2., 0.3), (di1., 0.9), (di2., 0.7)\}$. The two first rules express that for each disease we have an appropriate drug, these two drugs are incompatible, but their appropriateness is fully certain. The third (resp. fourth) rule expresses that drug $dr1$ (resp. $dr2$) has an efficiency quasi certain (resp little certain) for the disease $di1$ (resp $di2$). The two last rules (some observations) represent that the diagnosis of the disease $di1$ (resp. $di2$) is quasi certain (resp. almost certain). Then, the two possibilistic stable models of P are $\{(di1, 0.9), (di2, 0.7), (dr1, 0.9), (c1, 0.7)\}$ and $\{(di1, 0.9), (di2, 0.7), (dr2, 0.7), (c2, 0.3)\}$. So, the doctor* *can observe that he has an alternative. On one hand, he can give the drug $dr1$ and the patient will be almost certainly cured of the disease $di1$. On the other hand, he can give the drug $dr2$ and the patient will be cured of the disease $di2$ with a low certainty degree. However, if the doctor considers that disease $di2$ is very serious, maybe he will choose the drug $dr2$ even if the degree is lower. That is why it is interesting to keep the two stable models in which every conclusion is weighted with a certainty degree.*

Proposition 3.1 *Let P be a p.n.l.p.*

1. *Let A be a possibilistic stable model of P and $\alpha \in \mathcal{N}, \alpha > 0$, then $(x, \alpha) \in A \iff \alpha = N_{PA^*}(x)$.*
2. *Let S be a stable model of P^*, then $\{(x, N_{PS}(x)) \mid x \in \mathcal{X}, N_{PS}(x) > 0\}$ is a possibilistic stable model of P.*
3. *Let A be a possibilistic stable model of P, then A^* is a stable model of P^*.*

This result shows that there is a one-to-one mapping between the possibilistic stable models of a p.n.l.p P and the stable models of its classical part P^*. It shows that the decision problem of existence of a p.s.m. for a p.n.l.p. stays in the same complexity class as the decision problem of existence of a stable model for a normal logic program. So, it is an NP-complete problem. Furthermore, it indicates an easy way to implement a system able to compute a p.s.m. of a p.n.l.p P. First, one can use a software (as Smodels [Niemelä and Simons, 1997] or DLV [Eiter *et al.*, 1998]) to compute the stable models of P^*. Second, for every found stable model A, our operator ΠT_{PA} can be used to compute (in polynomial time) the corresponding p.s.m. $\Pi Cn(P^A)$ of P.

3.2 Possibility Distribution defined upon a Normal Logic Program

In the previous definition 3.2 we have proposed a syntactic way to compute the possibilistic stable models of a p.n.l.p. by using the fix-point operator ΠCn defined for a p.d.l.p. Now, we examine the semantics that can be given to this framework by defining a possibility distribution induced by the necessity values associated to each normal rules. This distribution $\tilde{\pi}$ over all the atom sets (i.e. over $2^{\mathcal{X}}$) has to reflect the ability of every atom set to be a stable model of P^* and that is achieved by the next definitions.

Definition 3.3 *Let P be a p.n.l.p. and A be an atom set, then $\tilde{\pi}_P$ is the possibility distribution defined by $\tilde{\pi}_P : 2^{\mathcal{X}} \rightarrow [0, 1]$ and respecting: $\forall A \in 2^{\mathcal{X}}, \tilde{\pi}_P(A) = \pi_{PA}(A)$.*

One can observe that definition of $\tilde{\pi}_P$ can be paraphrased by: "the possibility for an atom set A to be a stable model of P is its possibility to be the least model of the program P reducted by A". This definition is natural wrt. the definition 1.1 of a stable model and is justified by the next result.

Proposition 3.2 *Let P be a p.n.l.p. and $A \in 2^{\mathcal{X}}$ be an atom set, then $\tilde{\pi}_P(A) = 1 \iff A$ is a stable model of P^*.*

Example 3.2 *The possibility distribution induced by the p.n.l.p. $P = \{(a., 1), (b., 1), (c \leftarrow a, not\ d., 0.4), (d \leftarrow b, not\ c., 0.8), (e \leftarrow c., 0.6), (e \leftarrow d., 0.5)\}$ is null for every atom set included in $\{a, b, c, d, e\}$ except for $\tilde{\pi}_P(\{a, b\}) = 0.2$, $\tilde{\pi}_P(\{a, b, c\}) = 0.4$, $\tilde{\pi}_P(\{a, b, d\}) = 0.5$ and $\tilde{\pi}_P(\{a, b, c, e\}) = \tilde{\pi}_P(\{a, b, d, e\}) = 1$ ($\{a, b, c, e\}$ and $\{a, b, d, e\}$ are the two stable models of P^*).*

Definition 3.4 *Let P be a p.n.l.p. and $\tilde{\pi}_P$ its associated possibility distribution, we define the two dual possibility and necessity measures such that:*

$$\tilde{\Pi}_P(x) = \max_{A \in 2^{\mathcal{X}}} \{\tilde{\pi}_P(A) \mid x \in A\}$$
$$\tilde{N}_P(x) = 1 - \max_{A \in 2^{\mathcal{X}}} \{\tilde{\pi}_P(A) \mid x \notin A\}.$$

Proposition 3.3 *Let P be a consistent p.n.l.p. and $x \in \mathcal{X}$,*

1. *x is a credulous consequence of $P^* \Rightarrow \tilde{\Pi}_P(x) = 1$*
2. *x is not a skeptical consequence of $P^* \iff \tilde{N}_P(x) = 0$*

Let us remark that if an atom x is not a credulous consequence of P then it does not necessarily imply that $\tilde{\Pi}_P(x) = 0$ as it is the case for an atom that is not a consequence of a p.d.l.p. (see proposition 2.3 item 1). For instance, $P = \{(a., 0.6), (b \leftarrow not\ a., 0.7)\}$ has only one p.s.m. $\{(a, 0.6)\}$ so b is not a possibilistic credulous consequence of P. But, the possibility distribution is $\tilde{\pi}_P(\emptyset) = 0.3, \tilde{\pi}_P(\{a\}) = 1, \tilde{\pi}_P(\{b\}) = 0.4$ and $\tilde{\pi}_P(\{a, b\}) = 0$ so $\tilde{\Pi}_P(b) = 0.4$. This is because, with this program, b is not completely impossible. In fact, b is not a credulous consequence only because a blocks the applicability of the rule concluding b. So, in other words if we have a then we can not have b, but if a is absent then we can have b. Since, the certainty of a is only 0.6, so the possibility of b is naturally 0.4.

4 Discussion and Conclusion

In this work, we have proposed a way to deal with logic programs taking into account a level of certainty for each rule. First, we have given two equivalent ways, a semantic and a syntactic one, for dealing with possibilistic definite logic programs. Then, we have shown how to extend our framework to possibilistic normal logic programs in order to obtain what we called possibilistic stable models. In addition to this work, we have described in [Nicolas *et al.*, 2005] an application of our proposal to deal with inconsistent normal logic programs.

There are alternative formalisms for extending non-monotonic reasoning with qualitative or quantitative weights. In [Nerode *et al.*, 1997], annotated non-monotonic rule systems are introduced to affect on every piece of knowledge a weight representing a probability, an uncertainty measure, a time, a place,... For instance for a logic program, it leads to rules like $(c, 0.4) \leftarrow (a, 0.5), (not\ b, 0.7)$. This approach is different to our since we put weights on rules and not on each atom in the rules. Another divergence is the meaning of the weights : in our work we deal with uncertainty while they develop a formalism in which the semantic of weights is not established a priori. In [Damasio and Pereira, 2001a; 2001b; Dubois *et al.*, 1991; Benferhat *et al.*, 1993; Loyer and Straccia, 2003; Alsinet and Godo, 2000] the reader can find different propositions about multi-valued or probabilistic logic programs, about possibilistic definite logic programs, about levels of certainty ranking atoms, but not rules, involved in a non-monotonic reasoning,... But, none of these works describes a formalism dealing with uncertainty in a logic program with default negation by means of possibilistic theory, both in a semantic and a syntactic ways. The closest work of ours can be found in [Wagner, 1997]. But the approach is to reconstruct possibilistic logic by defining a multi-valued interpretation and a corresponding satisfaction relation and forgetting the notion of possibility distribution. Then, he is not able to provide any result about the possibility of an atom set to be a stable model. Moreover, the necessity (certainty) of a conclusion is not correlated to the possibility of the atom set. So, we think that this approach is far from the spirit of possibilistic logic.

In possibilistic logic, the necessity degrees are commonly interpreted as preferences between formulas, the more certain is a formula, the more preferred it is. Since there is a lot of works dealing with preferences between rules in non-monotonic reasoning [Delgrande *et al.*, 2004], it is interesting to analyze our work when we consider that the necessity degrees on rules determine a preference order. If we look only on the area of ASP, most of these works use the preferences expressed between the rules to make a choice between the different stable models to keep only the "preferred" ones. In other words, the priorities between the rules do not evaluate a certainty degree of the rules but are a tool to choice between contradicting rules. This differs from our work because, when several stable models exist, we compute the certainty of the propositions with respect to each stable model. But we do not try to eliminate some stable models since we consider they are alternate solutions. Nevertheless, a future work analyzing in depth the links between our proposal and the preference handling in ASP is envisaged, in particular with respect to the two general principles stated in [Brewka and Eiter, 1999].

Some direct, and without theoretical difficulties, continuation of this work is to precise the handling of strong negation in possibilistic normal programs. Then, we also plan to develop a system to compute possibilistic stable models by interfacing it with existing systems able to compute classical stable models. Furthermore, we will try to extend our approach to disjunctive logic programs and answer set semantics [Gelfond and Lifschitz, 1991].

References

[Alsinet and Godo, 2000] T. Alsinet and L. Godo. A complete calculus for possibilistic logic programming with fuzzy propositional variables. In C. Boutilier and M. Goldszmidt, editors, *UAI '00: Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence*, pages 1–10, Stanford University, Stanford, California, USA, 2000. Morgan Kaufmann.

[Benferhat *et al.*, 1993] S. Benferhat, D. Dubois, and H. Prade. Possibilistic logic: From nonmonotonicity to logic programming. In M. Clarke, R. Kruse, and S. Moral, editors, *European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 747 of *LNCS*, pages 17–24. Springer, 1993.

[Brewka and Eiter, 1999] G. Brewka and T. Eiter. Preferred answer sets for extended logic programs. *Artificial Intelligence*, 109(1-2):297–356, 1999.

[Damasio and Pereira, 2001a] C. V. Damasio and L. M. Pereira. Antitonic logic programs. In T. Eiter, W. Faber, and M. Truszczynski, editors, *International Conference on Logic Programming and NonMonotonic Reasoning*, volume 2173 of *LNCS*, pages 379–392. Springer, 2001.

[Damasio and Pereira, 2001b] C. V. Damasio and L. M. Pereira. Monotonic and residuated logic programs. In S. Benferhat and P. Besnard, editors, *European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 2143 of *LNCS*, pages 748–759. Springer, 2001.

[Delgrande *et al.*, 2004] J. Delgrande, T. Schaub, H. Tompits, and K. Wang. A classification and survey of preference handling approaches in nonmonotonic reasoning. *Computational Intelligence*, 20(2):308–334, 2004.

[Dubois and Prade, 1998] D. Dubois and H. Prade. Possibility theory: qualitative and quantitative aspects. In Ph. Smets, editor, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 1, pages 169–226. Kluwer Academic Press, 1998.

[Dubois *et al.*, 1991] D. Dubois, J. Lang, and H. Prade. Towards possibilistic logic programming. In K. Furukawa, editor, *International Conference on Logic Programming*, pages 581–595, Cambridge, MA, 1991. MIT Press.

[Dubois *et al.*, 1995] D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 439–513. Oxford University Press, 1995.

[Eiter *et al.*, 1998] T. Eiter, N. Leone, C. Mateis, G. Pfeifer, and F. Scarcello. The kr system dlv:progress report, comparisons and benchmarks. pages 406–417. Morgan Kaufmann Publishers, 1998.

[Gelfond and Lifschitz, 1988] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In Robert A. Kowalski and Kenneth Bowen, editors, *International Conference on Logic Programming*, pages 1070–1080. The MIT Press, 1988.

[Gelfond and Lifschitz, 1991] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3-4):363–385, 1991.

[Loyer and Straccia, 2003] Y. Loyer and U. Straccia. Default knowledge in logic programs with uncertainty. In *International Conference on Logic Programming*, volume 2916 of *LNCS*, pages 466–480, Mumbai, India, 2003. Springer.

[Nerode *et al.*, 1997] A. Nerode, J. B. Remmel, and V. S. Subrahmanian. Annotated nonmonotonic rule systems. *Theoretical Computer Science*, 171(1-2):111–146, 1997.

[Nicolas *et al.*, 2005] P. Nicolas, L. Garcia, and I. Stéphan. A possibilistic inconsistency handling in answer set programming. In *European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, LNCS, Barcelona, Spain, 2005. Springer.

[Niemelä and Simons, 1997] I. Niemelä and P. Simons. Smodels - an implementation of the stable model and well-founded semantics for normal logic programs. In *International Conference on Logic Programming and NonMonotonic Reasoning*, volume 1265 of *LNCS*, pages 420–429. Springer, 1997.

[Wagner, 1997] G. Wagner. A logical reconstruction of fuzzy inference in databases and logic programs. In *Proc. of Int. Fuzzy Set Association World Congress (IFSA)*, Prague, Czech, 1997.

[Zadeh, 1978] L.A. Zadeh. Fuzzy sets as a basis for a theory of possibility. In *Fuzzy Sets and Systems*, volume 1, pages 3–28. 1978.