Possibilistic Safe Beliefs vs. Possibilistic Stable Models

Ruben Octavio Velez Salazar, Jose Arrazola Ramirez, and Ivan Martinez Ruiz

Benemérita Universidad Autónoma de Puebla, Mexico

ruvelsa@yahoo.com

Abstract. In this paper we show an application of possibilistic stable models to a learning situation. Our main result is that possibilistic stable models of possibilistic normal programs are also possibilistic safe beliefs of such programs. In any learning process, the learners arrive with their previous knowledge. In most cases, it is incomplete or it comes with some degree of uncertainty. Possibilistic Logic was developed as an approach to automated reasoning from uncertain or prioritized incomplete information. The standard possibilistic expressions are classical logic formulas associated with weights. Logic Programming is a very important tool in Artificial Intelligence. Safe beliefs were introduced to study properties and notions of answer sets and Logic Programming from a more general point of view. The stable model semantics is a declarative semantics for logic programs with default negation. In [1], the authors present possibilistic safe beliefs. In [2], the authors introduce possibilistic stable models.

Keywords. Normal logic programs, safe beliefs, possibilistic logic, possibilistic normal logic programs, possibilistic safe beliefs.

1 Introduction

In the mid 80's Dubois and Prade [3] introduced Possibilistic Logic, a logic initially based on classical logic, useful for modeling problems where incomplete or partially contradictory information exists. It deals with uncertainty in the following way: in order to express the extent to which the available evidence entails the truth of a formula which is associated to a number between 0 and 1 called its degree of necessity (or its certainty). If we wish to express the extent to which the truth of the formula is not incompatible with the available evidence we may use a degree of possibility. In this paper we will refer only to the degree of necessity of the formula φ, which is denoted by $n(\varphi)$.

Answer Set Programming is a form of declarative programming based on the stable semantics of logic programming. The definition of answer sets for augmented programs (which are a general type of programs) is based on finding minimal models of some reduced logic programs. Stable model semantics [4] is an answer set semantics for logic programs with default negation.

In [2], the authors use possibility theory to extend the non-monotonic semantics of stable models for logic programs with default negation. They define a clear semantics for such programs by introducing possibilistic stable models, taking into account a certainty level associated with each piece of knowledge.

Any logic whose set of provable formulas lies between intuitionistic and classical logic (inclusive) is known as an Intermediate logic. These logics are able to distinguish between a and $\neg\neg a$, a property which makes these logics suitable to characterize notions of logic programming. Pearce [5] established a link between Answer Set Programming and Intermediate Logics. The authors in [6], present an extension of answer sets, called safe beliefs, which they define based on intuitionistic logic and following ideas found in [5]. Their definition formalizes the idea that non monotonic inference can be achieved determining some formulas that one can *safely believe*.

In [1], the authors develop possibilisitc safe beliefs in order to broaden the scope of applications. They present a characterization of possibilistic safe beliefs in terms of possibilistic intuitionistic logic.

2 Background

In this section we first introduce the syntax of logic formulas considered in this paper. Then we present a few basic definitions of how logics can be built to interpret the meaning of such formulas in order to finally give a brief introduction to the logics that are relevant for the results of our later sections.

2.1 Syntax of Formulas

We consider a formal (propositional) language built from: an enumerable set L_0 of elements called *atoms* (denoted a, b, c, ...); the binary connectives $\wedge$ (*conjunction*), $\vee$ (*disjunction*) and $\rightarrow$ (*implication*); and the unary connective $\neg$ (*default negation*). Formulas (denoted φ, ψ, γ, ...) are constructed as usual by combining these basic connectives together.

We also use $\varphi \leftrightarrow \psi$ to abbreviate $(\varphi\rightarrow\psi) \wedge (\psi\rightarrow\varphi)$ and, following the tradition in logic programming, $\varphi\leftarrow\psi$ as an alternate way of writing $\psi\rightarrow\varphi$. A *theory* is just a set of formulas and, in this paper, we only consider finite theories. Moreover, if T is a theory, we use the notation L_T to stand for the set of atoms that occur in the theory T.

2.2 Logic Systems

We consider a *logic* simply as a set of formulas that satisfies the following two properties: (i) is closed under modus ponens (i.e. if φ and $\varphi \rightarrow \psi$ are in the logic, then also ψ is) and (ii) is closed under substitution (i.e. if a formula φ is in the logic, then any other formula obtained by replacing all occurrences of an atom a in φ with another formula ψ is still in the logic). The elements of a logic are called *theorems* and the notation $\vdash_X \varphi$ is used to state that the formula φ is a theorem of the logic X

(i.e. $\varphi \in X$). We say that a logic X is *weaker than or equal to* a logic Y if $X \subseteq Y$, similarly we say that X is *stronger than or equal to* Y if $Y \subseteq X$.

Hilbert style proof systems. There are many different approaches that have been used to specify the meaning of logic formulas or, in other words, to define *logics* [7]. In Hilbert style proof systems, also known as axiomatic systems, a logic is specified by giving a set of axioms (which is usually assumed to be closed by substitution). This set of axioms specifies, so to speak, the "kernel" of the logic. The actual logic is obtained when this "kernel" is closed with respect to the inference rule of modus ponens.

The notation $\vdash_X \varphi$ for provability of a logic formula φ in the logic X is usually extended within Hilbert style systems, given a theory T, using $T \vdash_X \varphi$ to denote the fact that the formula φ can be derived from the axioms of the logic and the formulas contained in T by a sequence of applications of modus ponens.

2.3 Intuitionistic Logic

In this subsection we will briefly introduce the intuitionistic logic that will be relevant for our purposes in this paper. We will present a Hilbert style definition for it. We start from a basic logic called *Positive Logic*, to which we add some axioms in order to obtain Intuitionistic Logic.

Definition 1. Positive Logic *is defined by the following set of axioms:*

Pos 1: $\varphi \to (\psi \to \varphi)$
Pos 2: $(\varphi \to (\psi \to \gamma)) \to ((\varphi \to \psi) \to (\varphi \to \gamma))$
Pos 3: $\varphi \wedge \psi \to \varphi$
Pos 4: $\varphi \wedge \psi \to \psi$
Pos 5: $\varphi \to (\psi \to (\varphi \wedge \psi))$
Pos 6: $\varphi \to (\varphi \vee \psi$
Pos 7: $\psi \to (\varphi \vee \psi)$
Pos 8: $(\varphi \to \gamma) \to ((\psi \to \gamma) \to (\varphi \vee \psi \to \gamma))$

Definition 2. Intuitionistic Logic *I is defined as Positive Logic plus the following two axioms:*

Int1: $(\varphi\to \psi) \to [(\varphi\to \neg \psi) \to \neg \varphi]$
Int2: $\neg \varphi \to (\varphi\to \psi)$

2.4 Normal Logic Programs

In this paper, when a non empty set of atoms that determines the language of the programs is not given explicitly, we will consider it to be the set of atoms occurring in the program, so if P denotes the program then L_P denotes the set of atoms under consideration. A normal logic program is a finite set of formulas, called rules, of the form $r = c \leftarrow a_1,\ldots,a_n,\neg b_1,\ldots,\neg b_m$, where $n \geq 0$, $m \geq 0$, and $\{c,a_1,\ldots,a_n,b_1,\ldots,b_m\} \subseteq L_P$.

For such a rule r we use the following notations: $body^+(r) = \{a_1,...,a_n\}$, $body^-(r) = \{b_1,...,b_m\}$, $head(r) = c$, and $r^+ = head(r) \leftarrow body^+(r)$.

2.5 Possibilistic Logic

The standard possibilistic expressions of possibilistic logic are classical logic formulas associated with a parameter, interpreted as lower bounds of necessity degrees.

A *necessity-valued* formula is a pair (φ α), where φ is a propositional formula in some given logic and $\alpha \in (0,1]$. (φ α) expresses that φ is certain to the extent α, that is, $N(\varphi) \geq \alpha$, where N is a necessity measure which models our state of knowledge. The constant α is known as the *valuation* of the formula or its *weight* and is represented as $val(\varphi)$.

To define a possibilistic logic axiomatically, we start with a logic X (in section 4, X will be the Intuitionistic logic, which we denote by I) and define an axiom system by the set of axioms $\{(\varphi\ 1) : \varphi$ is an axiom of $X\}$ with the following rules of inference:

(GMP) $(\varphi\ \alpha), (\varphi \rightarrow \psi\ \beta) \vdash_{PXL} (\psi\ \min\{\alpha,\beta\})$.
(S) $(\varphi\ \alpha) \vdash_{PXL} (\varphi\ \beta)$ if $\alpha \geq \beta$.

This defines the possibilistic logic *PXL*. Let us point out that if $(\varphi\ \alpha) \vdash_{PXL} (\varphi\ \beta)$, then $\alpha \geq \beta$, so in the case $\beta > \alpha$ will not be considered. The reasoning behind (GMP) is the principle that the strength of a conclusion is the strength of the weakest argument used in its proof. Observe that every axiom and every theorem are associated with the value 1.

The following lemma can be found in [8] and we will use it in our main result.

Lemma 1. Let Γ be a set of formulas in *PXL* and $\varphi \in X$. Then $\Gamma \vdash_{PXL} (\varphi\ 1)$ if and only if $\Gamma^* \vdash_X \varphi$, where Γ^* is the collection of formulas in Γ without the corresponding parameters.

3 Possibilistic Stable Semantics

The stable model semantics was proposed in [4] for logic programs with default negation, i.e., normal logic programs. In order to deal with a reasoning which is non-monotonic and uncertain, the authors in [2] presented possibilistic stable models for possiblistic normal programs. We reproduce some of the main results here.

3.1 Possibilistic Definite Logic Programs

A *possibilistic definite (logic) program* is a set of possibilistic rules of the form

$$r = (c \leftarrow a_1,...,a_n\ \alpha),$$

where $n \geq 0$, $\{c, a_1, \ldots, a_n\} \subseteq L_P$, and $\alpha \in (0,1]$. The *classical projection* of the possibilistic rule is r* = $c \leftarrow a_1, \ldots, a_n$. The *weight* α is less than or equal to $n(r)$, the necessity degree representing the certainty level of the information described by r. If R is a set of possibilistic rules, then $R^* = \{r^* : r \in R\}$ is the definite logic program obtained by ignoring all the weights. By $A \not\models r^*$, we denote that r^* is not a logical consequence of the set of formulas A.

Definition 3.[2] *Let P be a possibilistic definite program.*

– *If M denotes the least Herbrand model of the definite program P*, then the* ***necessity measure*** *of an atom* $x \in L_P$ *is*

$$N_P(x) = \min_{A \subseteq M}\{\max_{r \in P}\{n(r) : A \not\models r^*\} : x \notin A\}.$$

– *The set* $\{(x\ N_P(x)) : x \in L_P, N_P(x) > 0\}$ *is* ***the possibilistic model*** *of P.*

$N_P(x)$ evaluates the level at which x is inferred from P. Moreover, $N_P(x)=0$ if and only if x does not belong to the least Herbrand model of the definite program P^*, hence the definition of the possibilistic model of P.

3.2 Possibilistic Normal Logic Programs

Normal Logic Programs allow default negation, as opposed to Definite Logic Programs, in which all the information described is positive. A *possibilistic normal (logic) program* is a finite set of rules of the form

$$r = (c \leftarrow a_1, \ldots, a_n, \neg b_1, \ldots, \neg b_m\ \alpha),$$

where $n \geq 0$, $m \geq 0$, $\{c, a_1, \ldots, a_n, b_1, \ldots, b_m\} \subseteq L_P$, and $\alpha \in (0,1]$.

In [4], the authors define the stable model semantics for normal logic programs in terms of a program reduction. This reduction is extended naturally to the possibilistic case as follows.

Definition 4. [2] *Let P be a possibilistic normal program.*

– *Let* $A \subseteq L_{P^*}$. *The* ***possibilistic reduct*** *of P with respect to A, which we denote by* P^A, *is the set*

$$\{((r^*)^+\ n(r)) : r \in P, body^-(r) \cap A = \emptyset\}.$$

– *Let* $M \subseteq L_P$. *M is a* ***possibilistic stable model*** *of P if M is the possibilistic model of* P^{M^*}.

With the following lemma, the authors in [2] show that there is a one-to-one mapping between the possibilistic stable models of a possibilistic normal logic program P and the stable models of its proyection P^*. We will use this fact in our main result.

Lemma 2. *Let P be a possibilistic normal program and* $M \subseteq L_P$. *If M is a possibilistic stable model of P then M* is a stable model of P*.*

We will also use this result, found in [5].

Lemma 3. *Let P be a logic program, $M \subseteq L_P$ and X be an intermediate logic. M is a stable model of P if and only if* $P \cup \neg\widetilde{M} \vdash_X M$.

In the next section, we will show that the possibilistic stable models of this possibilistic normal program are also its possibilistic safe beliefs.

4 Possibilistic Safe Beliefs

In [9], the authors present *safe beliefs* as an extension of answer sets in terms of completions of a program. In [1], the authors extend this notion to the possibilistic case. We reproduce some of their findings in this section, in which a *possibilistic theory* is a finite set of possibilistic formulas.

Definition 5. [1] *Let Γ be a possibilistic theory.*

– *The* ***inconsistency degree*** *of Γ, denoted as Incon(Γ), is defined as the following number:*

$$\max\{\alpha : \Gamma \vdash_{PIL} (\perp\ \alpha)\}.$$

– *Γ is* ***consistent*** *if* $Incon(\Gamma)=0$.

We note that we need to extend the domain of α from (0,1] to [0,1]. If *M* is a set of possibilistic atoms, we write $\widetilde{M}^*$ to denote the complement of M^* in L_{Γ^*}. Also, $\neg\neg M^*$ denotes the set $\{\neg\neg x : x \in M^*\}$ and $\Gamma \Vdash_{PIL} (\varphi\ \alpha)$ denotes that Γ is consistent and $\Gamma \vdash_{PIL} (\varphi\ \alpha)$.

If M is any subset of L_Γ we denote by $(\neg\widetilde{M}^*\ 1)$ and $(\neg\neg M^*\ 1)$ the sets $\{(x\ 1) : x \in \neg\widetilde{M}^*\}$ and $\{(x\ 1) : x \in \neg\neg M^*\}$, respectively.

It is possible to define a partial order in 2^{L_Γ} by defining for every M_1 and M_2 in 2^{L_Γ}, that $M_1 \leq M_2$ if the following two conditions hold:

a) $M_1 \subseteq M_2$;
b) If $(\varphi\ \alpha_1) \in M_1$ then there exists $(\varphi\ \alpha_2) \in M_2$ such that $\alpha_2 \leq \alpha_1$.

Definition 6. [1] *Let Γ be a possibilistic theory and M a subset of L_Γ. We define M to be a* ***possibilistic safe belief*** *of P if the following conditions are met:*

– *M is ≤ - minimal;*
– *For every* $(a\ \alpha) \in M$, $\Gamma \cup (\neg\widetilde{M}^*\ 1) \cup (\neg\neg\widetilde{M}^*\ 1) \Vdash_{PIL} (a\ \alpha)$.

The following lemma gives us the characterization we will use in our main result.

Lemma 4. [10] *Let P be a possibilistic normal program and $M \subseteq L_P$. M is a possibilistic safe belief of P if and only if* $P \cup (\neg\widetilde{M}^*\ 1) \vdash_{PIL} M$.

5 Contribution

Our main result follows from the previous lemmas.

Theorem 1. *Let P be a possibilistic normal program and* $M \subseteq L_P$. *If M is a possibilistic stable model of P then M is a possibilistic safe belief of P.*

Proof. If *M* is a possibilistic stable model of *P* then, by lemma 2, M* is a stable model of P*, which is equivalent, by lemma 3, to the fact that $P^* \cup \neg \widetilde{M}^* \vdash_I M^*$. Now, by lemma 1, we have $P \cup (\neg \widetilde{M}^*\ 1) \vdash_{PIL} M$, and therefore, by lemma 4, *M* is a possibilistic safe belief of *P*.

□The converse of theorem 1 does not hold: it is not difficult to verify that if *P* is the possibilistic normal program defined by the possibilistic rules ($a \leftarrow \neg b$ 0.5) and ($b \leftarrow \neg a$ 0.5), and if $M=\{(a\ 0.3)\}$ then *M* is a possibilistic safe belief of *P*, but not a possibilistic stable model of *P*.

6 Our Result and Learning Environments

We start this section with an example derived from [2]. Suppose a certain teacher has a student who has a hard time focusing on more than one subject. The teacher wishes to give the student a Math assignment and a History assignment, for each of which the student has some previous knowledge. The problem is that the student can only focus on Math or on History, but not both. We can represent this situation with a normal logic program

$$\{a \leftarrow b \wedge \neg c,\ c \leftarrow d \wedge \neg a,\ e \leftarrow a \wedge b,\ f \leftarrow c \wedge d,\ b \leftarrow,\ d \leftarrow\},$$

where the atoms *a* and *c* represent, respectively, the fact that the student uses his previous knowledge in Math and History; *b* and *d* represent, respectively, the fact that the student must complete the Math and History assignment; and *e* and *f* represent the fact that the student completes his assignment described by *b* and *d*, respectively. This normal program has two stable models, $\{a,b,d,e\}$ and $\{b,c,d,f\}$. Each one of these two stable models represents an option for the teacher, who now wishes to evaluate the *certainty* of these two options. In order to do so, the teacher uses her expertise, experience, etc. to determine degrees of certainty of each rule in the program. Now, the task is to figure out how the certainty of the rules in the program affect the certainty of the option described in each stable model.

After determining the degrees of certainty of each rule in the normal program

$$\{a \leftarrow b \wedge \neg c,\ c \leftarrow d \wedge \neg a,\ e \leftarrow a \wedge b,\ f \leftarrow c \wedge d,\ b \leftarrow,\ d \leftarrow\},$$

the teacher comes up with the possibilistic normal program

$$P=\{(a \leftarrow b \wedge \neg c\ 1),(c \leftarrow d \wedge \neg a\ 1),(e \leftarrow a \wedge b\ 0.7),(f \leftarrow c \wedge d\ 0.3),(b \leftarrow\ 0.9),(d \leftarrow\ 0.7)\}.$$

So she finds the necessity measures for each atom in the previous stable models, which result in two possibilistic stable models for *P*:

$M_1 = \{(a\ 0.9), (b\ 0.9), (d\ 0.7), (e\ 0.7)\}$ and $M_2 = \{(b\ 0.9), (c\ 0.7), (d\ 0.7), (f\ 0.3)\}$.

M_1 tells the teacher that she can give the student the Math assignment and the student almost certainly completes it. The certainty degree of the student completing his History assignment is much less. So now, the teacher may consider other factors, such as the students learning styles, in order to prioritize her options.

To end this section, let us reconsider the possibilistic normal program

$P=\{(a\leftarrow b \wedge \neg c\ 1),(c\leftarrow d \wedge \neg a\ 1),(e\leftarrow a \wedge b\ 0.7),(f\leftarrow c \wedge d\ 0.3),(b\leftarrow\ 0.9),(d\leftarrow\ 0.7)\}$

and its possibilistic stable models

$M_1 = \{(a\ 0.9), (b\ 0.9), (d\ 0.7), (e\ 0.7)\}$ and $M_2 = \{(b\ 0.9), (c\ 0.7), (d\ 0.7), (f\ 0.3)\}$.

It is not difficult to verify that $P \cup (\neg \widetilde{M}_1^*\ 1) \vdash_{PIL} M_1$ and that $P \cup (\neg \widetilde{M}_2^*\ 1) \vdash_{PIL} M_2$. Hence M_1 and M_2 are also possibilistic safe beliefs for P.

7 Future Work

Since lemma 3 applies to any logic program, not just normal logic programs, and lemma 4 applies to any possibilistic theory, we believe that our result may be extended to possibilistic disjunctive logic programs.

References

1. Estrada O., Arrazola J., Osorio M.: Possibilistic Safe Beliefs. LANMR 2010 (2010)
2. Nicolas P., Garcia L., Stephan I., Lefevre C.: Possibilistic uncertainty handling for answer set programming. Annals of Mathematics and Artificial Intelligence, Springer (2006)
3. Dubois D., Lang J., Prade H.: Possibilistic Logic. In: Gabbay D, Hogger C, Robinson J. (eds.) Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3, Clarendon Press Oxford (1994)
4. Gelfond M., Lifschitz V.: The Stable Model Semantics for Logic Programming. Fifth Conference on Logic Programming, MIT Press (1988)
5. Pierce D.: Stable Inference as Intuitionistic Validity. Logic Programming, 38 (1999)
6. Osorio M, Navarro J., Arrazola J.: Applications of Intuitionistic Logic in Answer Set Programming. Theory and Practice of Logic Programming (2004)
7. Mendelson E., Introduction to Mathematical Logic, CRC Press, Fifth Edition (2010)
8. Velez R., Arrazola J., Martinez, I.: Semantics for Some Non-Classical Possibilistic Logic. Under revision for publishing (2013)
9. Osorio M., Navarro J., Arrazola J.: Safe Beliefs for Propositional Theories. Annals of Pure and Applied Logic, Elsevier (2004)
10. Estrada E., Arrazola J., Osorio, M.: Possibilistic Intermediate Logic, International Journal of Advanced Intelligence Paradigms (IJAIP), Vol. 4, No. 2 (2012)