Extending Answer Set Programming using Generalized Possibilistic Logic

Didier Dubois
Université Paul Sabatier, CNRS
IRIT, Toulouse, France
dubois@irit.fr

Henri Prade
Université Paul Sabatier, CNRS
IRIT, Toulouse, France
prade@irit.fr

Steven Schockaert
Cardiff University
United Kingdom
SchockaertS1@cardiff.ac.uk

Abstract

Answer set programming (ASP) is a form of logic programming in which negation-as-failure is defined in a purely declarative way, based on the notion of a stable model. This short paper briefly explains how a recent generalization of possibilistic logic (GPL) can be used to characterize the semantics of answer set programming. This characterization has several advantages over existing characterizations of the stable model semantics. First, unlike reduct-based approaches, it does not rely on a syntactic procedure: we can directly characterize answer sets based on the minimally specific models of a GPL theory. Second, GPL enables us to study extensions of ASP in an intuitive way: unlike in existing generalizations of ASP such as equilibrium logic and autoepistemic logic, all formulas in GPL have a meaning which is intuitively clear. Finally, being based on possibilistic logic, GPL offers a natural way of dealing with uncertainty in answer set programs.

1 Introduction

An answer set program is a set of rules of the form:

$$a_1 \vee ... \vee a_n \leftarrow b_1 \wedge ... \wedge b_m \wedge \text{not}\, c_1 \wedge ... \wedge \text{not}\, c_r \quad (1)$$

where $a_1, ..., a_n, b_1, ..., b_m, c_1, ..., c_r$ are propositional literals, i.e. either atomic propositions from a given finite set *At* or the negation of such atomic propositions. We call $a_1 \vee ... \vee a_n$ the head of the rule while $b_1 \wedge ... \wedge b_m \wedge \text{not}\, c_1 \wedge ... \wedge \text{not}\, c_r$ is called the body. An extended literal is a literal or an expression of the form not c_i (with c_i a literal). Intuitively, (1) states that if we cannot derive that any of $c_1, ..., c_r$ are true and we can derive that all of $b_1, ..., b_m$ are true, then we should assume that one of $a_1, ..., a_n$ must be true. Formally, the semantics of an answer set program is defined in terms of the Gelfond-Lifschitz reduct [10]. In particular, given a set of literals M, the reduct P^M of an answer set program P is defined as follows:

$$P^M = \{a_1 \vee ... \vee a_n \leftarrow b_1 \wedge ... \wedge b_m \mid M \cap \{c_1, ..., c_r\} = \emptyset,$$
$$(a_1 \vee ... \vee a_n \leftarrow b_1 \wedge ... \wedge b_m \wedge \text{not}\, c_1 \wedge ... \wedge \text{not}\, c_r) \in P\}$$

We say that M is a model of P^M if for each rule $a_1 \vee ... \vee a_n \leftarrow b_1 \wedge ... \wedge b_m$ in P^M such that $\{b_1, ..., b_m\} \subseteq M$ it holds that $M \cap \{a_1, ..., a_n\} \neq \emptyset$. We say that M is an answer set of P^M if M is a model of P^M and there does not exist a model M' of P^M such that $M' \subset M$. Intuitively, the condition not c_i is satisfied if c_i cannot be derived from the program. However, what literals can be derived depends on which assumptions we make about what other literals can be derived, which introduces a circular dependency. When using the Gelfond-Lifschitz reduct, this dependency is broken by making a guess M about what literals can be derived, and then verifying that M indeed coincides with the set of literals that can be derived.

A large number of equivalent characterizations of answer sets have been proposed [15]. For example, autoepistemic logic (AEL [16]), the logic of minimal belief and negation as failure (MBNF [14]) and equilibrium logic (EL [18]) can be used to define answer sets in a model-theoretic way. Moreover, MBNF and EL can be used to define answer sets of arbitrary propositional combinations of extended literals (e.g. containing disjunctions of rules, negation as failure in the head of rules, etc.).

In this paper, we show how a recent generalization of possibilistic logic (GPL [5; 7; 8]) can be used to characterize answer sets. This characterization has several advantages over existing characterizations, in particular w.r.t. how it enables us to extend ASP. For example, GPL has the advantage over MBNF and EL that its models can be naturally interpreted as the epistemic state of an agent, which allows us to give an intuitive interpretation to answer set programs in which not c_1 means that "the agent does not know that c_1 is true". As a result, even when the syntax of ASP is extended to arbitrary propositional combinations of extended literals, the corresponding GPL formulas still have an intuitive meaning. In contrast, the intuitive meaning of EL formulas is not always clear. Moreover, since every propositional formula in GPL is encapsulated by a modal operator (see Section 2), we can distinguish between situations in which "the agent knows that either a or b holds" from "either the agent knows a or the agent knows b", whereas EL can only model the latter. The GPL characterization also ensures that all answer sets are minimal, i.e. that there cannot be two answer sets M_1 and M_2 such that $M_1 \subset M_2$. While this is true for any characterization of ASP when only rules of the form (1) are consid-

ered, existing characterizations do not guarantee minimality when negation-as-failure is allowed in the head of rules [13]. Finally, as the semantics of GPL is based on possibility distributions, the proposed characterization can be naturally extended to give a semantics to answer set programs in which rules can have uncertain conclusions.

In the next section, we briefly recall the syntax and semantics of GPL. Section 3 then explains how answer sets can be characterized using GPL, and how GPL can be used to define extensions of ASP. Finally, in Section 4 we consider uncertain answer set programs, and show how GPL can be used to define the possibilistic answer sets of such programs.

2 Generalized possibilistic logic

Let $\Lambda_k = \{0, \frac{1}{k}, ..., 1\}$ be the considered set of certainty degrees and let $\Lambda_k^+ = \Lambda_k \setminus \{0\}$. GPL formulas are then defined as follows:

- If $\lambda \in \Lambda_k^+$ and α is a propositional formula, then $\mathbf{N}_\lambda(\alpha)$ is a GPL formula;
- if α and β are GPL formulas, then so are $\neg\alpha$ and $\alpha \wedge \beta$.

As usual, we use $\alpha \rightarrow \beta$ and $\alpha \vee \beta$ as abbreviations for $\neg(\alpha \wedge \neg\beta)$ and $\neg(\neg\alpha \wedge \neg\beta)$. Furthermore we write $\mathbf{\Pi}_\lambda(\alpha)$ as an abbreviation for $\neg\mathbf{N}_{n(\lambda)}(\neg\alpha)$, where $n(\lambda) = 1 - \lambda + \frac{1}{k}$ for $\lambda \in \Lambda_k^+$. GPL is useful to reason about the knowledge of another agent. Intuitively $\mathbf{N}_\lambda(\alpha)$ means that the agent knows α with certainty λ, while $\mathbf{\Pi}_\lambda(\alpha)$ means that the agent considers α possible to the degree λ. An expression of the form $\mathbf{N}_\lambda(\alpha)$ or $\mathbf{\Pi}_\lambda(\alpha)$ will be called a meta-literal.

The semantics of GPL is defined in terms of possibility distributions. Let π be a normalized possibility distribution over the set of possible worlds Ω, i.e. π is a mapping from Ω to $[0, 1]$ such that $\pi(\omega) = 1$ for at least one ω in Ω. Then π is said to satisfy the GPL formula $\mathbf{N}_\lambda(\alpha)$, written $\pi \models \mathbf{N}_\lambda(\alpha)$, iff

$$N(\alpha) = \min\{1 - \pi(\omega) \mid \omega \in \Omega, \omega \models \neg\alpha\} \geq \lambda$$

where N denotes the necessity measure from possibility theory. The satisfaction relation $\models$ is extended to arbitrary GPL formulas in the usual way, i.e. $\pi \models \alpha \wedge \beta$ iff $\pi \models \alpha$ and $\pi \models \beta$, while $\pi \models \neg\alpha$ iff $\pi \not\models \alpha$. A possibility distribution π is called a model of a set of GPL formulas Θ if π satisfies every formula in Θ. An axiomatization of GPL has been presented in [7].

Let π_1 and π_2 be two possibility distributions over Ω. We say that π_1 is less specific than π_2, written $\pi_1 \preceq \pi_2$, if $\pi_1(\omega) \geq \pi_2(\omega)$ for all $\omega \in \Omega$. If $\pi_1 \preceq \pi_2$ and $\pi_1 \neq \pi_2$, we write $\pi_1 \prec \pi_2$. We say that π is a minimally specific model of a set of GPL formulas Θ if π is a model of Θ and there is no model π' of Θ such that $\pi' \prec \pi$. Let α be a GPL formula and let Θ be a set of GPL formulas. The following three inference relations are considered in GPL:

standard entailment $\Theta \models \alpha$ iff α is satisfied by every model of Θ.

cautious entailment $\Theta \models^c \alpha$ iff α is satisfied by all minimally specific models of Θ.

brave entailment $\Theta \models^b \alpha$ iff α is satisfied by at least one minimally specific model of Θ.

The problems of checking whether $\Theta \models \alpha$, $\Theta \models^c \alpha$ and $\Theta \models^b \alpha$ hold are respectively coNP-complete, Π_2^P-complete and Σ_2^P-complete [8].

GPL generalizes standard possibilistic logic [4; 6] in that the latter only allows sets of meta-literals of the form $\mathbf{N}_\lambda(\alpha)$, which are usually written as weighted propositional formulas of the form (α, λ). At the semantic level, a theory in possibilistic logic corresponds to a single possibility distribution, which is the unique minimally specific model of that theory, whereas theories in GPL can have several minimally specific models.

3 Characterizing and extending ASP using GPL

Given an answer set program P, we let Θ_P be the GPL theory which contains for each rule of the form (1) in P the following formula:

$$\mathbf{N}_1(b_1) \wedge ... \wedge \mathbf{N}_1(b_m) \wedge \mathbf{\Pi}_1(\neg c_1) \wedge ... \wedge \mathbf{\Pi}_1(\neg c_r) \rightarrow \mathbf{N}_1(a_1) \vee ... \vee \mathbf{N}_1(a_n) \quad (2)$$

In other words, the body of a rule of the form (1) is satisfied if the agent knows each b_i with maximal certainty and moreover the agent considers $\neg c_j$ fully possible for each j. Note that $\mathbf{\Pi}_1(\neg c_j)$ is equivalent to $\neg\mathbf{N}_{\frac{1}{k}}(c_j)$.

The transformation in (2) is by itself not enough, as ASP is based on the idea of forward chaining and in particular does not allow contrapositive reasoning (e.g. from the rule $a \leftarrow b$ and the fact $\neg a$ we should not derive $\neg b$). To see how forward chaining could be enforced using GPL, first note that there are three ways in which the formula (2) can be satisfied by a minimally specific model π of Θ_P:

1. one of the meta-literals $\mathbf{N}_1(b_i)$ is not satisfied by π;
2. one of the meta-literals $\mathbf{\Pi}_1(c_i)$ is not satisfied by π, i.e. $\mathbf{N}_{\frac{1}{k}}(c_i)$ is satisfied by π;
3. one of the meta-literals $\mathbf{N}_1(a_i)$ is satisfied by π.

The first case intuitively corresponds to an answer set which does not include b_i, i.e. to a situation in which the rule (1) does not apply. The third case intuitively corresponds to an answer set in which a_i has been included to make the rule (1) satisfied, i.e. to a situation in which a_i has been derived using (non-deterministic) forward chaining. The second case, however, intuitively corresponds to a contrapositive inference, i.e. (1) has been satisfied by making c_i true. The latter inference is not allowed in ASP and the second case should thus be excluded. To this end, we take advantage of the fact that it is only in the second case that certainty degrees other than 0 or 1 are needed. Note that here we do not use degrees for modelling uncertainty, but intuitively for differentiating between literals that are assumed to be true and literals that can effectively be derived. In particular, it turns out that answer sets correspond to the minimally specific models of Θ_P in which only the certainty degrees 0 and 1 occur. Formally, the requirement that only these certainty degrees occur is encoded

by the GPL formula Φ, defined as follows:

$$\Phi \equiv \bigwedge_{a \in At} \mathbf{N}_1(a) \vee \mathbf{N}_1(\neg a) \vee (\mathbf{\Pi}_1(a) \wedge \mathbf{\Pi}_1(\neg a)) \quad (3)$$

The formula Φ expresses that for every atom a, the agent is either fully certain about the truth value of a (in which case $\mathbf{N}_1(a) \vee \mathbf{N}_1(\neg a)$ holds) or the agent is completely ignorant about a (in which case $\mathbf{\Pi}_1(a) \wedge \mathbf{\Pi}_1(\neg a)$ holds). It turns out that the answer sets of P correspond to the minimally specific models of Θ_P that satisfy Φ. In particular, assuming that $k \geq 2$ (i.e. $|\Lambda_k| \geq 3$), it holds that P has a consistent answer set iff

$$\Theta_P \models^b \Phi$$

Moreover, for each consistent answer set M of P, it holds that the following possibility distribution π_M is a minimally specific model of Θ_P which satisfies Φ:

$$\pi_M(\omega) = \begin{cases} 1 & \text{if } \omega \text{ satisfies every literal in } M \\ 0 & \text{otherwise} \end{cases}$$

Conversely, for every minimally specific model π of Θ_P which satisfies Φ, it holds that the following set of literals M_π is an answer set of P:

$$M_\pi = \{l \,|\, N(l) = 1\}$$

where N is the necessity measure induced by π. Note that it follows that a literal l is included in at least one answer set of P iff

$$\Theta_P \models^b \Phi \wedge \mathbf{N}_1(l)$$

and that that l is included in all answer sets of P iff

$$\Theta_P \models^c \Phi \rightarrow \mathbf{N}_1(l)$$

This means in particular that the main reasoning tasks from ASP correspond to the standard forms of GPL inference.

This characterization of answer sets in GPL can be straightforwardly generalized to arbitrary propositional combinations of extended literals. When negation-as-failure in the head is considered, however, our characterization deviates from the existing characterizations in terms of EL and MBNF. This is illustrated in the next example.

Example 1. *We consider a single atom a, in which case $\Omega = \{\emptyset, \{a\}\}$, where we identify models with the set of atoms they make true. Consider the formula $a \vee$ not a, and the corresponding GPL encoding $\mathbf{N}_1(a) \vee \mathbf{\Pi}_1(\neg a)$. Clearly, the latter formula has a unique minimally specific model π, defined by $\pi(\emptyset) = \pi(\{a\}) = 1$. In other words, the only answer set we find for $a \vee$ not a is the empty set. However, both the characterization of ASP in MBNF [14] and the characterization in EL [18] find two answer sets for this example, viz. $M_1 = \emptyset$ and $M_2 = \{a\}$.*

As the intuition behind the stable model semantics is based on the idea of minimal commitment, the GPL semantics appears more natural.

The use of the modal operators $\mathbf{N}_\lambda$ in GPL allows us to further extend ASP. In the standard semantics of ASP, rules with disjunctions in the head intuitively correspond to a non-deterministic choice, e.g. $a \vee b \leftarrow c$ means that when the agent knows c then either it knows a or it knows b. When modelling epistemic reasoning, however, it often seems more natural to interpret $a \vee b$ as "the agent knows that either a or b is true (but may not know which is the case)". This latter reading was referred to as weak disjunction in [2], where the inference problems resulting from interpreting ASP rules in this way have been investigated. Using GPL, the ASP rule $a \vee b \leftarrow c$ can be modelled as $\mathbf{N}_1(c) \rightarrow \mathbf{N}_1(a \vee b)$ when weak disjunction is considered, and as $\mathbf{N}_1(c) \rightarrow \mathbf{N}_1(a) \vee \mathbf{N}_1(b)$ otherwise.

Finally, the use of possibilistic logic enables us to consider uncertain answer set programs. This is discussed in more detail in the next section.

4 Modelling uncertain answer set programs

An uncertain ASP rule is an expression of the form

$$\lambda : a_1 \vee ... \vee a_n \leftarrow b_1 \wedge ... \wedge b_m \wedge \text{not } c_1 \wedge ... \wedge \text{not } c_r \quad (4)$$

where λ is a certainty degree from Λ_k^+. An uncertain answer set program is a set of uncertain ASP rules. As has been observed in [1], there are two fundamentally different ways to interpret uncertain ASP rules. On the one hand, we may view λ as reflecting the certainty that the corresponding ASP rule is valid. This interpretation leads us to view an uncertain ASP program as a possibility distribution over classical ASP programs; at the semantic level, we can then consider a possibility distribution over classical answer sets. On the other hand, we may view λ as reflecting the certainty with which we can derive the head of the rule, given that its body is satisfied. This view leads to a semantics in which answer sets correspond to weighted epistemic states, which are modelled as possibility distributions. Note that a similar distinction is often made in first-order probabilistic logics [11] and in first-order conditional logics [9]. The former interpretation of uncertain ASP programs has been considered in [1] and [12]. The latter interpretation has been considered, among others, in [17], [2] and [3].

Modelling the former type of uncertain ASP programs in GPL would require nested modalities, encapsulating formulas of the form (2) with a modality of the form $\mathbf{N}_\lambda$. As nested modalities are not allowed in GPL, this would require us to define a variant of GPL in which every (standard) GPL formula would be encapsulated by a modality, similar to how propositional formulas are encapsulated in standard GPL. At the semantic level, models would then correspond to possibility distributions over possibility distributions over possible worlds.

Here we focus on modelling the second type of uncertain ASP programs in GPL. Let us first consider rules without negation-as-failure:

$$\frac{l}{k} : a_1 \vee ... \vee a_n \leftarrow b_1 \wedge ... \wedge b_m$$

The corresponding GPL formula is given by

$$\bigwedge_{i=1}^{l} \left(\mathbf{N}_{\frac{i}{k}}(b_1) \wedge ... \wedge \mathbf{N}_{\frac{i}{k}}(b_m) \rightarrow \mathbf{N}_{\frac{i}{k}}(a_1) \vee ... \vee \mathbf{N}_{\frac{i}{k}}(a_n) \right)$$

Let P be an uncertain ASP program without negation-as-failure and let Θ_P be the set of corresponding GPL formulas. It is not hard to see that a possibility distribution π is a possibilistic answer set of P in the sense of [2] iff π is a minimally specific model of Θ_P. In absence of negation-as-failure, the semantics from [1] moreover coincides with the semantics from [17].

To deal with negation-as-failure, [2] and [17] rely on a generalization of the Gelfond-Lifschitz reduct. Consider a rule of the following form:

$$\frac{l}{k} : a_1 \vee ... \vee a_n \leftarrow b_1 \wedge ... \wedge b_m \wedge \text{not}\, c_1 \wedge ... \wedge \text{not}\, c_r$$

The reduct of this rule w.r.t. a possibility distribution π, according to the semantics from [2], is given by:

$$\lambda : a_1 \vee ... \vee a_n \leftarrow b_1 \wedge ... \wedge b_m \tag{5}$$

where $\lambda = \min(\frac{l}{k}, \Pi(\neg c_1), ..., \Pi(\neg c_r))$, for Π the possibility measure induced by π. The reduct considered in [17] boils down to choosing $\lambda = \frac{l}{k}$ if $\Pi(\neg c_1) = ...\Pi(\neg c_r) = 1$ and $\lambda = 0$ otherwise (where rules whose certainty is 0 are simply omitted).

Using GPL, we can avoid the use of a reduct, if we assume that k is even and only certainty degrees from $\{\frac{2}{k}, \frac{4}{k}, ..., 1\}$ are used in the uncertain ASP program P. We can always ensure that this assumption is satisfied by replacing the set of certainty degrees Λ_k by the set Λ_{2k}, since every element from Λ_k is equal to $\frac{l}{2k}$ for some even value of l. The GPL theory Θ_P corresponding to P is then obtained by replacing every rule of the form (4) by the following formula:

$$\bigwedge_{i=1}^{l} \big(\mathbf{N}_{\frac{i}{k}}(b_1) \wedge ... \wedge \mathbf{N}_{\frac{i}{k}}(b_m) \wedge \mathbf{\Pi}_{\frac{i}{k}}(\neg c_1) \wedge ... \wedge \mathbf{\Pi}_{\frac{i}{k}}(\neg c_r) \rightarrow \mathbf{N}_{\frac{i}{k}}(a_1) \vee ... \vee \mathbf{N}_{\frac{i}{k}}(a_n)\big) \tag{6}$$

where we assume $\lambda = \frac{l}{k}$. We again need to exclude models which intuitively rely on contrapositive reasoning. Similar as in Section 3, these correspond to minimally specific models π which make (6) satisfied by making one of the meta-literals $\mathbf{\Pi}_{\frac{i}{k}}(\neg c_1)$ false, i.e. by making a meta-literal $\mathbf{N}_{\frac{k-i+1}{k}}(\ c_1)$ true. Noting that $k - i + 1$ is odd iff i is even, we find that it suffices to exclude models in which certainty degrees $\frac{l}{k}$ are used with l odd. Such models can be avoided by considering the following GPL formula Φ_k

$$\Phi_k \equiv \bigwedge_{a \in At} \bigwedge_{i=1}^{\frac{k}{2}} \big(\mathbf{N}_{\frac{2i-1}{k}}(a) \rightarrow \mathbf{N}_{\frac{2i}{k}}(a)) \wedge (\mathbf{N}_{\frac{2i-1}{k}}(\neg a) \rightarrow \mathbf{N}_{\frac{2i}{k}}(\neg a)\big)$$

Note that Φ_2 is equivalent to the formula Φ defined in (3). We propose the following definition: ϕ is a possibilistic answer set of P iff π is a minimally specific model of Θ_P which satisfies Φ_k. As in Section 3, we can then formulate the main reasoning tasks for possibilistic ASP in terms of standard entailment in GPL. For example, checking whether the certainty of a is at least $\frac{l}{k}$ (with l even) in every possibilistic answer set of P corresponds to:

$$\Theta_P \models^c \Phi_k \rightarrow \mathbf{N}_{\frac{l}{k}}(a)$$

We can show that the proposed definition of possibilistic answer set corresponds to the reduct-based definition from [2].

5 Conclusions

We have shown how generalized possibilistic logic (GPL) can be used to characterize answer sets without the need for a reduct operation, and how this characterization allows us to consider a range of different extensions of ASP in a natural way. In particular, the GPL characterization enables us to define answer sets for arbitrary propositional combinations of extended literals (while keeping the intuition of minimal commitment), for modelling weak disjunction, and for defining answer sets of uncertain programs.

Acknowledgment

Steven Schockaert has been supported by a grant from the Leverhulme Trust (RPG-2014-164).

References

[1] K. Bauters, S. Schockaert, M. De Cock, and D. Vermeir. Semantics for possibilistic answer set programs: Uncertain rules versus rules with uncertain conclusions. *International Journal of Approximate Reasoning*, 55(2):739–761, 2014.

[2] K. Bauters, S. Schockaert, M. De Cock, and D. Vermeir. Characterizing and extending answer set semantics using possibility theory. *Theory and Practice of Logic Programming*, 15(1):79–116, 2015.

[3] R. Confalonieri, J. C. Nieves, M. Osorio, and J. Vázquez-Salceda. Dealing with explicit preferences and uncertainty in answer set programming. *Annals of Mathematics and Artificial Intelligence*, 65(2-3):159–198, 2012.

[4] D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In D. N. D. Gabbay, C. Hogger J. Robinson, editor, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 439–513. Oxford University Press, 1994.

[5] D. Dubois and H. Prade. Generalized possibilistic logic. In *Proceedings of the 5th International Conference on Scalable Uncertainty Management*, pages 428–432, 2011.

[6] D. Dubois and H. Prade. Possibilistic logic - An overview. In D. M. Gabbay, J. Siekmann, and J. Woods, editors, *Handbook of the History of Logic, Vol. 9: Computational Logic*, pages 283–342. Elsevier, 2014.

[7] D. Dubois, H. Prade, and S. Schockaert. Stable models in generalized possibilistic logic. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 519–529, 2012.

[8] D. Dubois, H. Prade, and S. Schockaert. Reasoning about uncertainty and explicit ignorance in generalized possibilistic logic. In *Proceedings of the 21st European Conference on Artificial Intelligence*, pages 261–266, 2014.

[9] N. Friedman, J. Y. Halpern, and D. Koller. First-order conditional logic for default reasoning revisited. *ACM Transactions on Computational Logic*, 1(2):175–207, 2000.

[10] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proceedings of the Fifth International Conference and Symposium on Logic Programming*, pages 1081–1086, 1988.

[11] J. Y. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46(3):311 – 350, 1990.

[12] J. Hu, M. Westphal, and S. Wölfl. Towards a new semantics for possibilistic answer sets. In *Proceedings of the 37th Annual German Conference on AI*, pages 159–170. 2014.

[13] K. Inoue and C. Sakama. Negation as failure in the head. *The Journal of Logic Programming*, 35(1):39 – 78, 1998.

[14] V. Lifschitz. Minimal belief and negation as failure. *Artificial Intelligence*, 70(1–2):53 – 72, 1994.

[15] V. Lifschitz. Twelve definitions of a stable model. In *Proceedings of the 24th International Conference on Logic Programming*, pages 37–51, 2008.

[16] R. C. Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25(1):75 – 94, 1985.

[17] P. Nicolas, L. Garcia, I. Stéphan, and C. Lefèvre. Possibilistic uncertainty handling for answer set programming. *Annals of Mathematics and Artificial Intelligence*, 47:139–181, 2006.

[18] D. Pearce. Equilibrium logic. *Annals of Mathematics and Artificial Intelligence*, 47:3–41, 2006.