Minimal Undefinedness for Fuzzy Answer Sets

Mario Alviano  Giovanni Amendola  Rafael Peñaloza
Department of Mathematics and Computer Science  KRDB Research Centre
University of Calabria, Italy  Free University of Bozen-Bolzano, Italy
{alviano, amendola}@mat.unical.it  rafael.penaloza@unibz.it

Abstract
Fuzzy Answer Set Programming (FASP) combines the non-monotonic reasoning typical of Answer Set Programming with the capability of Fuzzy Logic to deal with imprecise information and paraconsistent reasoning. In the context of paraconsistent reasoning, the fundamental principle of minimal undefinedness states that truth degrees close to 0 and 1 should be preferred to those close to 0.5, to minimize the ambiguity of the scenario. The aim of this paper is to enforce such a principle in FASP through the minimization of a measure of undefinedness. Algorithms that minimize undefinedness of fuzzy answer sets are presented, and implemented.

Introduction
Fuzzy Answer Set Programming (FASP) (Nieuwenborgh, Cock, and Vermeir 2007; Janssen et al. 2012a; 2012b; Blondeel et al. 2013; Lee and Wang 2014; Mushthofa, Schölkopf, and Cock 2015) is a successfully combination of Fuzzy Logic (Cintula, Hájek, and Noguera 2011) and Answer Set Programming (ASP) (Gelfond and Lifschitz 1991; Marek and Truszczyński 1999; Niemelä 1999), which resulted in a declarative framework supporting non-monotonic reasoning on propositional formulas interpreted by truth degrees in the interval [0, 1]. As in ASP, reasoning on unknown knowledge is eased by the use of default negation, whose semantics is elegantly captured by the notion of answer set, or stable model: in a model, truth of unknown knowledge may be assumed as soon as there is no evidence of the contrary, and the model is discarded when the truth of some propositions is not necessary in order to satisfy the input program under the assumption for the unknown knowledge provided by the model itself. Moreover, as in Fuzzy Logic, non-Boolean truth degrees are useful to handle vague information, but also to deal with inconsistencies that may arise from mathematical abstractions of real phenomena. In this respect, the truth degree 0.5 is analogous to the truth value undefined used by many paraconsistent logics and paraconsistent answer set semantics (Przymusinski 1991; You and Yuan 1994; Sakama and Inoue 1995; Eiter, Leone, and Saccà 1997; Amendola et al. 2016).

Fuzzy answer sets satisfy two fundamental principles shared by several semantics for logic programs: justifiability (J, or foundedness), and the closed world assumption (C, or CWA). Briefly, for normal ASP programs, justifiability requires that every true atom is derived from the input program under the assumption provided for the negated formulas, and the CWA constrains to false any atom whose defining rules have false bodies (You and Yuan 1994; Eiter, Leone, and Saccà 1997). For FASP programs, these two principles can be recast in terms of truth degrees:

- (JC) Any truth degree \( x \in [0, 1] \) for an atom \( p \) is derived from the input program under the assumption provided for the negated formulas.

Minimal undefinedness (U) is another fundamental principle introduced in the context of paraconsistent reasoning, imposing a minimization on the number of undefined atoms (You and Yuan 1994). For example, the FASP program \( \Pi = \{ p \leftarrow \neg q, q \leftarrow \neg p \} \) has three answer set candidates, namely \( I = \{(p,1), (q,0)\} \), \( J = \{(p,0), (q,1)\} \), and \( K = \{(p,0.5), (q,0.5)\} \), but \( K \) has to be discarded because it contains two undefined atoms. In terms of truth degrees, minimal undefinedness imposes the following preference:

- (U) Truth degrees close to 0 or 1 should be preferred to those close to 0.5.

However, minimal undefinedness is not enforced by the current notion of fuzzy answer set, as for example \( K \) is among the fuzzy answer sets of the program \( \Pi \) above. In fact, \( \Pi \) has uncountably many fuzzy answer sets of the form \( \{(p,x), (q,1-x)\}, x \in [0, 1]. I \) and \( J \) should be the preferred two fuzzy answer sets for being undefinedness-free.

The previous example also highlights that fuzzy answer sets do not possess another important principle known as congruence (Co): The extension of a semantics must coincide with the original semantics on coherent theories. In terms of fuzzy ASP, congruence can be stated as follows:

- (Co) Fuzzy answer sets of coherent ASP programs coincide with crisp answer sets.

Principles (U) and (Co) are useful in abduction processes involving fuzzy circuits. For example, the designer of a fuzzy controller may be interested in computing input configurations producing a given output. This abduction process can be improved by focusing on minimal undefined interpretations, as those are the simplest to explain in the real world.
Example 1. Consider a fuzzy controller with temperature and humidity as input, and fan speed as output. Let $t_1, t_2, t_3$ and $h_1, h_2, h_3$ be atoms representing different classes of temperatures and humidities (e.g., $t_1$ is cool, $t_2$ is normal, and $t_3$ is warm; $h_1$ is humidity low, $h_2$ is humidity normal, and $h_3$ is humidity high) and $s_1, s_2, s_3$ represent different classes of fan speeds (e.g., $s_1$ is off, $s_2$ is normal speed, and $s_3$ is maximum speed). The fuzzy controller rules are $\Pi_3 = \{ s_1 \leftarrow t_1, s_2 \leftarrow t_2 \odot h_3, s_3 \leftarrow t_3 \odot (h_2 \odot h_1) \}$; i.e., the fan is turned off when the temperature is cool, set to normal speed when the temperature is normal and the humidity is high, and set to maximum speed when the temperature is warm and the humidity is normal or high. A possible input for the controller is $\Pi_{in} = \{ t_1 \leftarrow 0, t_2 \leftarrow 0.8, t_3 \leftarrow 0.2, h_1 \leftarrow 0, h_2 \leftarrow 0.1, h_3 \leftarrow 0.9 \}$, representing a slightly warm temperature and a high humidity. In this case the controller sets the fan speed to a value slightly higher than normal. Indeed, as will be clear after formally introducing the semantics in the next section, the truth degrees of $s_1, s_2, s_3$ are respectively 0, 0.7, and 0.2.

In this context, the designer of the fuzzy controller may be interested in checking the existence of input configurations such that all output variables $s_1, s_2, s_3$ are assigned the truth degree 0, hence leaving the fan speed completely undetermined. For example, in $\Pi_3$ this is the case when $h_1, t_3$ are 1, and all other input variables are 0.

In this paper the notion of fuzzy answer sets is refined by means of a measure of undefinability, which results into the definition of minimal undefinability fuzzy answer set. Principles (JC), (U), and (Co) are satisfied by FASP after this refinement. Algorithms to iteratively compute fuzzy answer sets that decrease the measure of undefinability are also presented, among them binary and progression search. The algorithms are implemented in a prototype system extending FASP2SMT (Alviano and Peñaloza 2015), a FASP solver using Z3 (de Moura and Bjørner 2008) as backend. The performance of these algorithms is compared, on the same solver, with the use of the MINIMIZE instruction of Z3.

Preliminaries

Let $\mathcal{B}$ be a fixed set of propositional variables. A fuzzy atom (atom for short) is either a variable from $\mathcal{B}$, or a numeric constant in the interval $[0,1]$. Fuzzy expressions are defined inductively as follows: every atom is a fuzzy expression; if $\alpha$ and $\beta$ are fuzzy expressions and $\odot \in \{ \odot, \oplus, \forall, \neg \}$ is a connective, then $\neg \alpha$ and $\alpha \odot \beta$ are fuzzy expressions, where $\neg$ denotes default negation. The connectives $\odot, \oplus$ are called Łukasiewicz, and $\forall, \neg$ are the Gödel connectives. A positive expression is a fuzzy expression not using $\neg$. A rule is of the form $\alpha \leftarrow \beta$, with $\alpha$ an atom, and $\beta$ a fuzzy expression. This rule is positive if $\beta$ is a positive expression. A (general) FASP program is a finite set of rules. A positive FASP program is a FASP program where every rule is positive. We use $\text{At}(\Pi)$ to denote the set of atoms occurring in $\Pi$.

The semantics of FASP programs requires a set of truth degrees. For the scope of this paper we consider the real interval $\mathcal{L}_0 = [0,1]$ and the sets $\mathcal{L}_n = \{ 1/n | 1 \leq n, \ldots, n-1 \}$, for $n \geq 2$. Note that $\mathcal{L}_2 = \{ 0, 1 \}$ is the classical Boolean set. Henceforth, $\mathcal{L}$ denotes an arbitrary but fixed such set, unless explicitly stated otherwise. A FASP program $\Pi$ is an $\mathcal{L}$-program if all constants occurring in $\Pi$ are in $\mathcal{L}$.

An $\mathcal{L}$-interpretation $I$ is a function $I : \mathcal{B} \rightarrow \mathcal{L}$ mapping each atom of $\mathcal{B}$ to a truth degree in $\mathcal{L}$. We will often denote such a function as the set $\{ (p, I(p)) \mid p \in \mathcal{B} \}$. $I$ is extended to fuzzy expressions as follows: $I(c) = c$ for all $c \in [0,1]$; $I(\neg \alpha) = 1 - I(\alpha)$; $I(\alpha \odot \beta) = \max(I(\alpha) + I(\beta) - 1, 0)$; $I(\alpha \oplus \beta) = \min(I(\alpha) + I(\beta), 1)$; $I(\alpha \forall \beta) = \max(I(\alpha), I(\beta))$; and $I(\alpha \neg \beta) = \min(I(\alpha), I(\beta))$. $I$ satisfies the rule $\alpha \leftarrow \beta$ if $I(\alpha) \geq I(\beta)$; it is an $\mathcal{L}$-model of an $\mathcal{L}$-program $\Pi$ if $I \models \Pi$ for all $\Pi \in \mathcal{L}$.

Given $\mathcal{L}$-interpretations $I, J$, we write $I \preceq J$ if $I(p) \leq J(p)$ for all $p \in \mathcal{B}$. If $I \preceq J$ and $I \neq J$, we write $I < J$. A minimal $\mathcal{L}$-model of an $\mathcal{L}$-program $\Pi$ is an $\mathcal{L}$-model of $\Pi$ such that there is no $\mathcal{L}$-model $J$ satisfying $J < I$. The reduct of an $\mathcal{L}$-program $\Pi$ w.r.t. an $\mathcal{L}$-interpretation $I$, denoted $\Pi^I$, is obtained by replacing each occurrence of $\neg \alpha$ by the constant $1 - I(\alpha)$. Let $\Pi$ be an $\mathcal{L}$-program, and $I$ be an $\mathcal{L}$-interpretation. $I$ is an $\mathcal{L}$-answer set of $\Pi$ if $I$ is a minimal $\mathcal{L}$-model of $\Pi$. FAS($\mathcal{L}, \Pi$) is the set of $\mathcal{L}$-answer sets of $\Pi$. $\Pi$ is $\mathcal{L}$-coherent if FAS($\mathcal{L}, \Pi$) $\neq \emptyset$, and $\mathcal{L}$-incoherent otherwise.

Example 2. Let $\Pi = \{ a \leftarrow b, b \leftarrow c, b \leftarrow 0.4 \}$, and $\Pi = \{ (a, 0), (b, 1), (c, 0) \}$. $I = \{ a \in \text{FAS}(\mathcal{L}, \Pi) \}$ because $I$ is a minimal $\mathcal{L}$-model of $\Pi^I = \{ a \leftarrow 0, b \leftarrow 1, b \leftarrow 0.4 \}$.

ASP programs can be seen as special cases of FASP 2-programs using only the connective $\forall$. The classical answer sets of such programs are FAS($\mathcal{L}_2, \Pi$).

Justifiability and CWA

Let us first introduce the immediate consequence operator $T_\Pi$ for a positive $\mathcal{L}$-program $\Pi$:

$$T_\Pi(I) : \alpha \mapsto \max\{ I(\beta) \mid \alpha \leftarrow \beta \in \Pi, \forall \alpha \in \mathcal{B} \}$$

where $I$ is an $\mathcal{L}$-interpretation. The operator is monotonic, and thus has at least a fixpoint $\text{lpf}(T_\Pi)$.

Given an $\mathcal{L}$-program $\Pi$ and an $\mathcal{L}$-interpretation $I$, an atom $p \in \mathcal{B}$ is justified in $\Pi$ and $I$ if $I(p) = \text{lpf}(T_\Pi)(p)$. Note that the immediate consequence operator is applied to the reduct $\Pi^I$, so that $\text{lpf}(T_{\Pi^I})(p)$ actually derives the truth degree of $p$ from the input program under the assumption provided by $I$ for the negated expressions in $\Pi$.

Theorem 1 (Justifiability). Let $\Pi$ be an $\mathcal{L}$-program, and $I \in \text{FAS}(\mathcal{L}, \Pi)$. All atoms $p \in \mathcal{B}$ such that $I(p) \in (0,1]$ are justified in $\Pi$ and $I$.

Proof. We show that $\text{lpf}(T_{\Pi^I})$ equals $I$. The reduct $\Pi^I$ has the unique $\mathcal{L}$-answer set $\text{lpf}(T_{\Pi^I})$ (Łukasiewicz 2006). Hence, there is no $J < \text{lpf}(T_{\Pi^I})$ with $J = (\Pi^I)^J$. Since $(\Pi^I)^J = \Pi^I$, we have that $\text{lpf}(T_{\Pi^I})$ is a minimal $\mathcal{L}$-model of $\Pi^I$. It follows that $\text{lpf}(T_{\Pi^I}) = I$ because $I \in \text{FAS}(\mathcal{L}, \Pi)$ by assumption.

Consequently, atoms not occurring in the left-hand side of any rule of $\Pi$, among those in $\mathcal{B}, \text{At}(\Pi)$, are associated with truth degree 0 in any $\mathcal{L}$-answer set of $\Pi$.

Corollary 1 (CWA). Let $\Pi$ be an $\mathcal{L}$-program, $p \in \mathcal{B}$, and $I \in \text{FAS}(\mathcal{L}, \Pi)$. If $I(\beta) = 0$ for all $p \leftarrow \beta \in \Pi$, then $I(p) = 0$. 

As stated in the introduction, justifiability and CWA are important properties of fuzzy answer sets, but insufficient to produce simple explanations in abductive reasoning. The next example clarifies this aspect.

**Example 3.** Let $\Pi_{ab}$ be $\Pi_c$ from Example 1 extended with the following rules:

\[
\begin{align*}
  & p \leftarrow \neg \neg p \quad \forall p \in \{t_1, t_2, t_3, h_1, h_2, h_3\} \quad (2) \\
  & 0 \leftarrow \neg (t_1 \oplus t_2 \oplus t_3) \quad (3) \\
  & 0 \leftarrow \neg (h_1 \oplus h_2 \oplus h_3) \quad (4) \\
  & 0 \leftarrow s_1 \oplus s_2 \oplus s_3 \quad (5)
\end{align*}
\]

Fuzzy answer sets of $\Pi_{ab}$ correspond to input configurations of the fuzzy controller in Example 1 such that all output variables are assigned 0. Indeed, (2) guesses an input configuration $I$, (3)–(4) enforce $I(t_1) + I(t_2) + I(t_3) \geq 1$ and $I(h_1) + I(h_2) + I(h_3) \geq 1$ (some restrictions are omitted for simplicity), and (5) discards $I$ if $I(s_1) + I(s_2) + I(s_3) > 0$. Two fuzzy answer sets of $\Pi_{ab}$ are $\{t_1, 0\}, \{t_2, 0.5\}, \{t_2, 0.5, h_1, 0.5\}, \{h_2, 0.5\}, \{h_1, 0\}, \{s_1, 0\}, \{s_2, 0\}, \{s_3, 0\}$ and $\{t_1, 0\}, \{t_2, 0\}, \{t_3, 1\}, \{h_1, 1\}, \{h_2, 0\}, \{h_1, 0\}, \{s_1, 0\}, \{s_2, 0\}, \{s_3, 0\}$. The latter is a simpler explanation, and should be preferred to the former. ■

### Minimal Undefined Fuzzy Answer Sets

Fuzzy answer sets can be seen as a kind of imprecise answer set, where the interpretation of some of the atoms may not be fully defined. We want to restrict our attention only to those fuzzy answer sets that minimize the undefinedness according to an appropriate measure. Following De Luca and Termini (1972), a measure of undefinedness is a function $U$ mapping every interpretation $I$ to a non-negative real number $U(I) \in \mathbb{R}$ such that:

- **(P1)** $U(I) = 0$ if and only if $I(p) \in \{0, 1\}$ for all $p \in \mathcal{B}$;
- **(P2)** if for every $p \in \mathcal{B}$ either (i) $J(p) \geq I(p) \geq 0.5$, or (ii) $J(p) \leq I(p) \leq 0.5$, then $U(I) \geq U(J)$; and
- **(P3)** let $\bar{I}(p) := 1 - I(p)$ for all $p \in \mathcal{B}$; then $U(I) = U(\bar{I})$.

Intuitively, these properties state, respectively, that classical interpretations are fully defined; interpretations closer to the extreme degrees are more defined (or less undefined); and undefinedness is symmetric w.r.t. complementary interpretations. Given a measure of undefinedness $U$, $U(J) < U(I)$ can be understood as $J$ being more informative than $I$. Hence, minimizing $U$ corresponds to selecting a fuzzy answer sets with minimal imprecision, and potentially taking only the extreme degrees 0 and 1. Note that the properties **(P1)**–**(P3)** imply that the interpretation mapping all variables to the intermediate value 0.5 will always maximize the value of $U$.

In some cases it is also interesting to consider measures of undefinedness in which $U$ increases strictly as the interpretations get farther from the borders, in order to satisfy the principle (U). Formally, a measure of undefinedness is **strict** if it satisfies the following property:

- **(P2)** if $|I(p) - 0.5| > |J(p) - 0.5|$ for all $p \in \mathcal{B}$, and for some $q \in \mathcal{B}$ $|I(q) - 0.5| > |J(q) - 0.5|$, then $U(I) < U(J)$.

A simple example of a (strict) measure of undefinedness is the distance function $U_D$ that measures how distant are the interpretations of the atoms from being classical. Formally,

\[
U_D(I) = \sum_{p \in \mathcal{B}} \min\{I(p), 1 - I(p)\}.
\]

**Theorem 2** ($U_D$). $U_D$ is a strict measure of undefinedness.

**Proof.** We need to show the properties (P1), (P2) and (P3).

- **(P1)** $U_D(I) = 0$ iff $\min\{I(p), 1 - I(p)\} = 0$ for all $p \in \mathcal{B}$, which holds iff $I(p) \in \{0, 1\}$ for all $p \in \mathcal{B}$.

- **(P2)** This property follows from the observation that $\min\{\alpha, 1 - \alpha\} = 0.5 - |\alpha - 0.5|$, for each $\alpha \in \mathbb{R}$. Given $I, J$, by assumption, $|I(p) - 0.5| \geq |J(p) - 0.5|$ for all $p \in \mathcal{B}$ and at least one of these inequalities is strict. Then,

\[
U_D(I) = \sum_{p \in \mathcal{B}} (0.5 - |I(p) - 0.5|) < \sum_{p \in \mathcal{B}} (0.5 - |J(p) - 0.5|) = U_D(J).
\]

- **(P3)** Define the sets $\mathcal{A}(I) = \{p \in \mathcal{B} \mid I(p) \geq 0.5\}$ and $\mathcal{A}'(I) = \{p \in \mathcal{B} \mid I(p) < 0.5\}$. It is easy to see that

\[
U_D(I) = \sum_{p \in \mathcal{A}(I)} (1 - I(p)) + \sum_{p \in \mathcal{A}'(I)} (I(p))
\]

\[
= \sum_{p \in \mathcal{A}(I)} (I(p)) + \sum_{p \in \mathcal{A}'(I)} (1 - I(p)) = U_D(\bar{I}). \square
\]

Notice, however, that many other such measures exist. A non-strict measure of undefinedness that can sometimes be considered is the drastic measure that maps crisp interpretations to 0 and all others to 1.

**Definition 1.** An $\mathcal{L}$-answer set $I$ of a program $\Pi$ is a minimal undefined $\mathcal{L}$-answer set w.r.t. the strict measure $U$ if there is no $J \in \text{FAS}(\mathcal{L}, \Pi)$ with $U(J) < U(I)$. The set of minimal undefined $\mathcal{L}$-answer sets of $\Pi$ w.r.t. $U$ is denoted by $\text{MUFAS}(\mathcal{L}, \Pi, U)$.

Clearly, $\text{MUFAS}(\mathcal{L}, \Pi, U) \subseteq \text{FAS}(\mathcal{L}, \Pi)$ holds. As shown in the following example, there are cases in which this inclusion is strict. Thus, restricting to minimally undefined interpretations further specializes the class of models of interest, satisfying property (U) given in the introduction.

**Example 4.** The interpretations $I = \{(a, 0.1), (b, 0.9)\}$ and $J = \{(a, 0.6), (b, 0.4)\}$ are two $\mathcal{L}_\infty$-answer sets of the program $\Pi = \{a \leftarrow \neg b, b \leftarrow \neg a\}$ with $U_D(I) < U_D(J)$. Indeed,

\[
U_D(I) = \min\{1 - I(a), I(a)\} + \min\{1 - I(b), I(b)\}
\]

\[
= 0.9 + 0.1 = 1.0,
\]

and

\[
U_D(J) = 0.4 + 0.4 = 0.8.
\]

Therefore, $J \notin \text{MUFAS}(\mathcal{L}_\infty, \Pi, U_D)$. On the other hand, also $I \notin \text{MUFAS}(\mathcal{L}_\infty, \Pi, U_D)$ holds, as it can easily be shown that $\text{MUFAS}(\mathcal{L}_\infty, \Pi, U_D) = \{(a, 0), (b, 1), (a, 1), (b, 0)\}$. ■

Observe that the program $\Pi$ in Example 4 satisfies $\text{MUFAS}(\mathcal{L}_\infty, \Pi, U_D) = \text{FAS}(\mathcal{L}_2, \Pi)$, that is, its minimal undefined $\mathcal{L}_\infty$-answer set coincides with its classical answer sets. This is not by chance, as claimed in the next theorem.
Theorem 3 (Congruence). If Π is an \( \mathcal{L}_2 \)-coherent program, then \( \text{MUFAS}(\mathcal{L}, \Pi, U) = \text{FAS}(\mathcal{L}_2, \Pi) \) for all sets of truth degrees \( \mathcal{L} \) and measures of undefinedness \( U \).

Proof. (\( \subseteq \)) Consider \( I \in \text{FAS}(\mathcal{L}_2, \Pi) \). Since \( U \) is a measure of undefinedness, by (P1), \( U(I) = 0 \). Therefore, there is no fuzzy interpretation \( J \) such that \( U(J) < U(I) \), i.e., \( I \in \text{MUFAS}(\mathcal{L}, \Pi, U) \).

(\( \supseteq \)) Let \( I \in \text{MUFAS}(\mathcal{L}, \Pi, U) \). We show that \( U(I) = 0 \), from which \( I \in \text{FAS}(\mathcal{L}_2, \Pi) \) follows by (P1). Since \( \Pi \) is \( \mathcal{L}_2 \)-coherent, there is \( J \in \text{FAS}(\mathcal{L}_2, \Pi) \), and hence \( U(J) = 0 \). Since \( J \in \text{FAS}(\mathcal{L}, \Pi) \) holds as well, we have \( U(I) = 0 \).

Notice that if \( \Pi \) is \( \mathcal{L}_2 \)-incoherent, then it may be the case that \( \text{MUFAS}(\mathcal{L}, \Pi, U) \neq \text{FAS}(\mathcal{L}_2, \Pi) \). For instance, the program \( \Pi = \{ a \leftarrow \neg a \} \) has no classical answer sets, but \( \{(a, 0.5)\} \) is in \( \text{MUFAS}(\mathcal{L}_2, \Pi, U) \).

Finally, one of the peculiarities of fuzzy answer set programming, in contrast to its classical version, is that a single FASP program may have uncountably many answer sets. As we show in the following example, the same holds for minimal undefined fuzzy answer sets.

Example 5. Let \( \Pi \) be \( \{ a \leftarrow \neg b \land c, b \leftarrow \neg a \lor c, c \leftarrow \neg c \} \). For every \( x \in [0, 0.5] \), \( I_x := \{(a, 0.5 - x), (b, x), (c, 0.5)\} \) is an \( \mathcal{L}_\infty \)-answer set of \( \Pi \) with \( U_D(I_x) = 1 \). Moreover, there is no \( J \in \text{FAS}(\mathcal{L}_\infty, \Pi) \) with \( U(J) < 1 \). Hence, \( \Pi \) has uncountably many minimal undefined fuzzy answer sets.

Algorithm 1: BinarySearch(\( \Pi, U, \varepsilon \))

1. \( (\text{sat}, I^*) := \text{solve}(\Pi) \);
2. if not sat then return incoherent;
3. \( (lb, ub) := (-\varepsilon, U(I^*)) \);
4. while \( ub - lb > \varepsilon \) do
   5. \( b := (lb + ub)/2 \);
   6. \( (sat, I) := \text{solve}(\Pi, U(\{(p, x, p) \mid p \in \text{Ar}(\Pi)\}) \leq b) \);
   7. if sat then \( (I^*, ub) := (I, U(I)) \);
   8. else \( lb := b \);
9. return \( I^* \);

This algorithm works by discarding any interpretation \( I \) such that \( U(I) > b \). Moreover, for an \( \mathcal{L}_\leq \)-program \( \Pi \) and a formula of the form (7), let \( \text{solve}(\Pi, \varphi) \) denote the invocation of a function computing a fuzzy answer set \( I \) of \( \Pi \) satisfying \( \varphi \), if it exists; in this case, the function returns \( \text{true, I} \), and otherwise it returns \( \text{false, -} \). Abusing of notation, we also use \( \text{solve}(\Pi) \) to invoke function \( \text{solve} \) on \( \Pi \) alone, with no bound on the measure of undefinedness.

The first algorithm we consider is based on binary search (Algorithm 1). Its input is an \( \mathcal{L}_\leq \)-program \( \Pi \) and a measure of undefinedness \( U \), and a precision threshold \( \varepsilon \in \mathbb{R}^+ \). The algorithm returns either the string incoherent, if \( \text{FAS}(\mathcal{L}_\leq, \Pi, U) = \emptyset \) (lines 1–2), or an \( \mathcal{L}_\leq \)-answer set \( I^* \in \text{FAS}(\mathcal{L}_\leq, \Pi) \) such that \( U(I^*) - U(I) < \varepsilon \) for all \( I \in \text{FAS}(\mathcal{L}_\leq, \Pi, U) \). If \( \Pi \) is coherent, the algorithm initializes a lower bound \( lb \) and an upper bound \( ub \) (line 3): the upper bound is the measure of undefinedness of the current solution, while the lower bound represents a measure of undefinedness that cannot be achieved by any \( \mathcal{L}_\leq \)-answer set of \( \Pi \). Then, the algorithm searches for an \( \mathcal{L}_\leq \)-answer set whose measure of undefinedness is at most \( (lb + ub)/2 \) (lines 5–6); if one is found, the current solution and the upper bound are updated (line 7), otherwise the lower bound is set to \( (lb + ub)/2 \); there is no \( I \in \text{FAS}(\mathcal{L}_\leq, \Pi) \) with \( U(I) \leq (lb + ub)/2 \) (line 8). The process is repeated until the possible improvement of the upper bound is below the given precision threshold \( \varepsilon \).

Example 6. Consider the program \( \Pi = \{ a \leftarrow \neg b, b \leftarrow \neg a \} \) from Example 4, with the measure \( U_D \), and \( \varepsilon = 0.1 \). solve(\( \Pi \)) returns a fuzzy answer set \( I_0 \) of \( \Pi \); say \( I_0 = \{(a, 0.6), (b, 0.4)\} \). Then \( lb \leftarrow -0.1 \) and \( ub \leftarrow 0.8 \). The algorithm then finds a new answer set \( I_1 \) with \( U(I_1) \leq 0.35 \); say \( I_1 = \{(a, 0.1), (b, 0.9)\} \), and updates the upper bound \( ub \leftarrow 0.2 \). At the next iteration we get \( I_2 = \{(a, 0.05), (b, 0.95)\} \), so \( ub \leftarrow 0.1 \). The algorithm sets \( b \leftarrow 0 \), and the answer set \( I_3 = \{(a, 1), (b, 0)\} \) is retrieved and returned.

The second method (Algorithm 2) uses progression search to find MUFAS. In this case, the undefinedness is bound to \( ub - pr \), where \( pr \) is the required improvement to the current solution. The variable \( pr \) is initially set to the precision threshold \( \varepsilon \) (line 3), doubled at each iteration (line 6), and reset to \( \varepsilon \) when it becomes too large (line 5).

Example 7. Using the input from Example 6, Algorithm 2 finds the fuzzy answer set \( I_0 = \{(a, 0.6), (b, 0.4)\} \), initializing \( (lb, ub, pr) \leftarrow (-0.1, 0.8, 0.1) \). It then searches for a new fuzzy answer set \( I_1 \) with \( U_D(I_1) \leq 0.7 \); say \( I_1 = \{(a, 0.3), \)}
\textbf{Algorithm 2: ProgressionSearch}(\Pi, U, \varepsilon)

1. (sat, \Pi') := \text{solve}(\Pi);
2. \textbf{if} not sat \textbf{then return} incoherent;
3. (lb, ub, pr) := (\varepsilon, \Pi', \varepsilon);
4. \textbf{while} ub - lb > \varepsilon \textbf{do}
5. \textbf{if} ub - pr \leq lb \textbf{then} pr := \varepsilon;
6. (b, pr) := (ub - pr, 2 \cdot pr);
7. (sat, I) := \text{solve}(\Pi, U, \{ (p, xp) \mid p \in \text{At}(\Pi) \} \leq b);
8. \textbf{if sat} \textbf{then} (I, ub) := (I, U(I));
9. \textbf{else} lb := b;
10. \textbf{return} I*;

(b, \{0.7\}), updating ub \leftarrow 0.6 and pr \leftarrow 0.2. The next iteration restricts answer sets to have a measure of at most 0.4, yielding e.g. I_2 = \{ (a, 0.1), (b, 0.9) \} and updating ub \leftarrow 0.2, pr \leftarrow 0.4. Since ub - pr \leq 0, pr is reset to 0.1, and a new solution with \Pi_0(I_2) \leq 0.1 is sought. The next iterations find I_3 = \{ (a, 0.05), (b, 0.95) \} and I_4 = \{ (a, 1), (b, 0) \}. At this point, ub - lb = \varepsilon, and I_4 is given as a solution.

Finally, we consider a third algorithm which we call \textit{e-improvement}. This method is obtained from Algorithm 1 by replacing line 5 to update b as follows:

5. \hspace{1cm} b := ub - \varepsilon

Intuitively, the modified algorithm minimally improves the measure of undefinedness of the current solution, until an incoherence arises.

\textbf{Example 8.} Consider the input from Example 6, yielding the first solution I_0 = \{ (a, 0.6), (b, 0.4) \} and lb \leftarrow 0.8. The next solution should have a measure of at most 0.7; hence, the algorithm retrieves e.g. I_1 = \{ (a, 0.3), (b, 0.7) \}. The next iteration, b is set to 0.5, which yields a new solution like I_2 = \{ (a, 0.1), (b, 0.9) \}. The new solution should then be bounded by 0.1. Thus, I_3 = \{ (a, 0.05), (b, 0.95) \} and I_4 = \{ (a, 1), (b, 0) \} are retrieved. The latter is returned.

\section*{Implementation and Evaluation}

The three algorithms presented above were implemented in a prototype system extending FASP2SMT (Alviano and Peñalosa 2015) with the distance function \Pi_0 as the measure of undefinedness. Other measures of undefinedness can be easily accommodated in the prototype, but left for future extensions. Briefly, FASP2SMT parses a symbolic \mathcal{L}\text{-program, which is then rewritten into ASP Core 2.0 (Alviano et al. 2013) and grounded by GRINGO (Gebser et al. 2011). The output of GRINGO encodes a propositional \mathcal{L}\text{-program} \Pi, which is parsed and rewritten into SMT-Lib (Barrett, Stump, and Tinel 2010), and processed by Z3 (de Moura and Björner 2008) to compute an \mathcal{L}\text{-answer set, if it exists.}

The new prototype keeps Z3 online, adds a formula of the form (6), and asks for a new model. Then, the formula (6) is removed, and this iteration is repeated until the desired precision threshold is reached. Besides the three algorithms presented here, FASP2SMT can use the minimize instruction of Z3, which however is experimental and not part of the SMT-Lib format. In this case z3 runs silently to completion, without intermediate solutions provided to the user.

The prototype system was tested empirically on satisfiable instances of Hamiltonian Cycle from the literature (Mushtofoh, Schockaert, and Cock 2014; Alviano and Peñalosa 2015). Each method was tested with threshold \varepsilon \in \{0.1, 0.01, 0.001\}, on an Intel Xeon CPU 2.4 GHz with 16 GB of RAM. Time and memory were limited to 600 seconds and 15 GB, respectively.

The results of the experiment are reported in Table 1. Binary search exhibits the best performance: it solves all instances if the required precision \varepsilon is set to 0.1, and anyhow the majority of the testcases if \varepsilon is smaller. The value of \varepsilon significantly affects the algorithm based on progression. Indeed, this algorithm reached the best performance when \varepsilon was set to 0.01. The larger value 0.01 resulted in several expensive invocations of function solve returning incoherent, while the smaller value 0.001 slowed down the search in some cases. Even if the algorithm is often unable to terminate, it can still provide to the user fuzzy answer sets with a good guarantee on the measure of undefinedness (the difference between lower and upper bound is usually lesser than 1). Algorithm \varepsilon\text{-improvement reports a very bad performance, with several timeouts and not negligible bound differences. Using the minimize construct of Z3 is not an option here, as its performance was similar to \varepsilon\text{-improvement.}

\section*{Related Work}

FAS and MUFAS are related to paracoherent answer set semantics. First we consider 3\text{-valued stable models semantics (one of the best known approximations of answer sets) introduced in (Przymusinska 1991). Each 3\text{-valued stable model (where false stands for 0, true for 1, and undefined for 0.5) corresponds to a fuzzy answer set. However, the reverse statement does not hold. For example, the only 3\text{-valued stable models of the program } \Pi \text{ from Example 4 are } \{ \{a, \sim b\}, \{a, b\}, \text{ and } \emptyset \text{, which correspond to } \{ (a, 1), (b, 0) \}, \{ (a, 0), (b, 1) \} \text{ and } \{ (a, 0.5), (b, 0.5) \}, \text{ respectively. On the other hand, } \Pi \text{ has infinitely many fuzzy answer sets. However, if we restrict to the } \mathcal{L}_3 \text{ semantics for programs using only the}}

<table>
<thead>
<tr>
<th>\varepsilon</th>
<th>solved</th>
<th>binary</th>
<th>progress</th>
<th>\varepsilon\text{-impr}</th>
<th>min.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100%</td>
<td>22%</td>
<td>0%</td>
<td>6%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.073</td>
<td>0.409</td>
<td>7.157</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>149</td>
<td>206</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>59</td>
<td>68</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>94%</td>
<td>67%</td>
<td>6%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.013</td>
<td>0.111</td>
<td>7.834</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>227</td>
<td>295</td>
<td>419</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>61</td>
<td>62</td>
<td>74</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>83%</td>
<td>44%</td>
<td>0%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.005</td>
<td>0.565</td>
<td>8.399</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>230</td>
<td>358</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>61</td>
<td>62</td>
<td></td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Gödel connectives, the 3-valued stable models and the class of fuzzy answer sets coincide. Moreover, Eiter, Leone, and Saccà (1997) note that 3-valued stable models leave more atoms undefined than necessary. Thus, they characterized 3-valued stable models in terms of Partial stable (P-stable) models and introduced the subclass of Least undefined-stable (L-stable) models. Intuitively, L-stable model semantics selects those 3-valued stable models, where a smallest set of atoms is undefined. Thus, MUFAS, restricted to the Gödel connectives, coincide to the L-stable models on FASP programs interpreted over $\mathcal{L}_3$. Finally, among other paraconsistent answer set semantics, we consider Semi-stable models (Sakama and Inoue 1995) and Semi-equilibrium models (Amendola et al. 2016). These paraconsistent semantics satisfy three desiderata (see Amendola et al. 2016): (i) every consistent answer set of a program corresponds to a paraconsistent model (answer set coverage); (ii) if a program has some (consistent) answer set, then its paraconsistent models correspond to answer sets (congruence); (iii) if a program has a classical model, then it has a paraconsistent model (classical coherence). In general, FAS semantics (and, thus, MUFAS semantics) does not satisfy this last property. For example, the program $\Pi =\{\neg a \leftarrow \neg b \land \neg c, a \leftarrow \neg b, b \leftarrow \neg c, c \leftarrow \neg a\}$ has no fuzzy answer set, while it has some classical model (for instance, setting $a$, $b$, and $c$ to true). Measure of undefinedness are often associated with the notion of entropy in information (Kapur and Kesavan 1992; Kullback 1959; Jayne 1957; Bhandari and Pal 1993; Li 2015; Li and Liu 2007; Wang, Dong, and Yan 2012), and applied in several areas: machine learning and decision trees (Qin, Li, and Ji 2009; Haber, del Toro, and Gajate 2010; Wang, Zhai, and Lu 2008; Yi, Lu, and Liu 2011; Zhai 2011); portfolio selection and optimization models (Qin, Li, and Ji 2009; Haber, del Toro, and Gajate 2010; Xie et al. 2010); clustering, image processing and computer vision (De Luca and Termini 1974; Yager 1979; 1980; Xie and Bedrosian 1984; Kosko 1986; Pal and Pal 1992; Shang and Jiang 1997; Szmidt and Kacprzyk 2001; Parkash, Sharma, and Mahajan 2008).

**Conclusions**

We have studied the notion of minimal undefinedness for fuzzy answer set programming as a means to identify solutions that satisfy additional desired properties. Intuitively, we are interested in solutions that are as close to being classical as possible. This study is motivated by previous work on paraconsistent and paraconsistent logical formalisms, but also by an attempt to enhance abduction processes in fuzzy circuits. More precisely, minimally undefined fuzzy answer sets yield the most precise, and easiest to understand explanations for an observed output.

Minimally undefined fuzzy answer sets, along with the measures of undefinedness used to define them, satisfy many properties that have been considered in the literature. In particular, they satisfy justifiability, the closed world assumption, and are coherent. Moreover, the distance function $U_D$ is a strict measure of undefinedness.

We implemented and evaluated four different methods for computing MUFAS based on the distance function, by extending the FASP2SMT system. Our evaluation shows that binary search provides the best strategy in this setting, while the internal (experimental) minimize instruction of Z3 yields the worst results. As future work we intend to extend the capabilities of our prototype to handle other measures of undefinedness and apply it to more realistic instances for abdication in fuzzy circuits. Moreover, the implementation can be improved by employing approximation operators (Alviano and Peñaloza 2013) to properly initialize lower bounds to values greater than $-e$. For example, for $\Pi =\{a \leftarrow 0.1 \lor \neg b, b \leftarrow 0.8 \land \neg a\}$ and $I \in \text{FAS}(\mathcal{L}_3,\Pi)$, $I(a) \in [0.1,0.2]$ and $I(b) \in [0.8,0.9]$ hold. Hence, $Ib$ in Algorithms 1–3 can be safely initialized to $0.2 - e$.

**References**


