Minimal Undefinedness for Fuzzy Answer Sets

Mario Alviano **Giovanni Amendola**
Department of Mathematics and Computer Science
University of Calabria, Italy
{alviano,amendola}@mat.unical.it

Rafael Peñaloza
KRDB Research Centre
Free University of Bozen-Bolzano, Italy
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, Schockaert, 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 paracoherent 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:

($\utilde{\mathbf{JC}}$) 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 $\Pi = \{p \leftarrow {\sim}q,\ q \leftarrow {\sim}p\}$ 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:

($\utilde{\mathbf{U}}$) 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 Π above. In fact, Π 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:

($\utilde{\mathbf{Co}}$) Fuzzy answer sets of coherent ASP programs coincide with crisp answer sets.

Principles ($\utilde{\mathbf{U}}$) and ($\utilde{\mathbf{Co}}$) 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* $\Pi_c = \{s_1 \leftarrow t_1,\ s_2 \leftarrow t_2 \otimes h_3,\ s_3 \leftarrow t_3 \otimes (h_2 \oplus h_3)\}$*; 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* Π_c *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 undefinedness, which results into the definition of minimal undefinedness fuzzy answer set. Principles $(\underset{\smile}{\mathbf{JC}})$, $(\underset{\smile}{\mathbf{U}})$, and $(\underset{\smile}{\mathbf{Co}})$ are satisfied by FASP after this refinement. Algorithms to iteratively compute fuzzy answer sets that decrease the measure of undefinedness 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 $\mathscr{B}$ be a fixed set of propositional variables. A *fuzzy atom* (*atom* for short) is either a variable from $\mathscr{B}$, or a numeric constant in the interval $[0,1]$. *Fuzzy expressions* are defined inductively as follows: every atom is a fuzzy expression; if α and β are fuzzy expressions and $\odot \in \{\otimes, \oplus, \veebar, \barwedge\}$ is a connective, then $\sim\alpha$ and $\alpha \odot \beta$ are fuzzy expressions, where $\sim$ denotes *default negation*. The connectives $\otimes, \oplus$ are called Łukasiewicz, and $\veebar, \barwedge$ are the Gödel connectives. A *positive expression* is a fuzzy expression not using $\sim$. A *rule* is of the form $\alpha \leftarrow \beta$, with α an atom, and β a fuzzy expression. This rule is *positive* if β 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 $At(\Pi)$ to denote the set of atoms occurring in Π.

The semantics of FASP programs requires a set of truth degrees. For the scope of this paper we consider the real interval $\mathscr{L}_\infty = [0,1]$ and the sets $\mathscr{L}_n = \{\frac{i}{n-1} | i = 0, ..., n-1\}$, for $n \geq 2$. Note that $\mathscr{L}_2 = \{0,1\}$ is the classical Boolean set. Henceforth, $\mathscr{L}$ denotes an arbitrary but fixed such set, unless explicitly stated otherwise. A FASP program Π is an *$\mathscr{L}$-program* if all constants occurring in Π are in $\mathscr{L}$. An *$\mathscr{L}$-interpretation* I is a function $I : \mathscr{B} \rightarrow \mathscr{L}$ mapping each atom of $\mathscr{B}$ into a truth degree in $\mathscr{L}$. We will often denote such a function as the set $\{(p, I(p)) \mid p \in \mathscr{B}\}$. I is extended to fuzzy expressions as follows: $I(c) = c$ for all $c \in [0,1]$; $I(\sim\alpha) = 1 - I(\alpha)$; $I(\alpha \otimes \beta) = \max\{I(\alpha) + I(\beta) - 1, 0\}$; $I(\alpha \oplus \beta) = \min\{I(\alpha) + I(\beta), 1\}$; $I(\alpha \veebar \beta) = \max\{I(\alpha), I(\beta)\}$; and $I(\alpha \barwedge \beta) = \min\{I(\alpha), I(\beta)\}$. I satisfies the rule $\alpha \leftarrow \beta$ ($I \models \alpha \leftarrow \beta$) if $I(\alpha) \geq I(\beta)$; it is an *$\mathscr{L}$-model* of an $\mathscr{L}$-program Π ($I \models \Pi$) if $I \models r$ for all $r \in \Pi$.

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

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

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

Justifiability and CWA

Let us first introduce the *immediate consequence operator* T_Π for a positive $\mathscr{L}$-program Π:

$$T_\Pi(I) : \alpha \mapsto \max\{I(\beta) \mid \alpha \leftarrow \beta \in \Pi\},\ \forall \alpha \in \mathscr{B} \quad (1)$$

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

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

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

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

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

Corollary 1 (CWA)**.** *Let* Π *be an* $\mathscr{L}$*-program,* $p \in \mathscr{B}$*, and* $I \in FAS(\mathscr{L}, \Pi)$*. 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 Π_{abd} be Π_c from Example 1 extended with the following rules:*

$$p \leftarrow {\sim}{\sim}p \qquad \forall p \in \{t_1, t_2, t_3, h_1, h_2, h_3\} \quad (2)$$
$$0 \leftarrow {\sim}(t_1 \oplus t_2 \oplus t_3) \quad (3)$$
$$0 \leftarrow {\sim}(h_1 \oplus h_2 \oplus h_3) \quad (4)$$
$$0 \leftarrow s_1 \oplus s_2 \oplus s_3 \quad (5)$$

Fuzzy answer sets of Π_{abd} 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 Π_{abd} are $\{(t_1,0),(t_2,0.5),(t_3,0.5),(h_1,0.5),(h_2,0.5),(h_3,0),(s_1,0),$ $(s_2,0),(s_3,0)\}$ and $\{(t_1,0),(t_2,0),(t_3,1),(h_1,1),(h_2,0),$ $(h_3,0),(s_1,0),(s_2,0),(s_3,0)\}$. 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 $p \in \mathscr{B}$;

(P2) if for every $p \in \mathscr{B}$ 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 $p \in \mathscr{B}$; 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 (Ṵ). Formally, a measure of undefinedness is *strict* if it satisfies the following property:

(P2') if $|I(p) - 0.5| \geq |J(p) - 0.5|$ for all $p \in \mathscr{B}$, and for some $q \in \mathscr{B}$ $|I(q) - 0.5| > |J(q) - 0.5|$, 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,

$$U_D(I) = \sum_{p \in \mathscr{B}} \min\{I(p), 1 - I(p)\}.$$

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 $p \in \mathscr{B}$, which holds iff $I(p) \in \{0,1\}$ for all $p \in \mathscr{B}$.

(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 $p \in \mathscr{B}$ and at least one of these inequalities is strict. Then,

$$\begin{aligned} U_D(I) &= \sum_{p \in \mathscr{B}} (0.5 - |I(p) - 0.5|) \\ &< \sum_{p \in \mathscr{B}} (0.5 - |J(p) - 0.5|) = U_D(J). \end{aligned}$$

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

$$\begin{aligned} U_D(I) &= \sum_{p \in \mathscr{A}(I)} (1 - I(p)) + \sum_{p \in \mathscr{A}'(I)} (I(p)) \\ &= \sum_{p \in \mathscr{A}(I)} (\bar{I}(p)) + \sum_{p \in \mathscr{A}'(I)} (1 - \bar{I}(p)) = U_D(\bar{I}). \; \square \end{aligned}$$

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

Clearly, $MUFAS(\mathscr{L}, \Pi, U) \subseteq FAS(\mathscr{L}, \Pi)$ 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 (Ṵ) 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 $\mathscr{L}_\infty$-answer sets of the program $\Pi = \{a \leftarrow {\sim}b, b \leftarrow {\sim}a\}$ with $U_D(I) < U_D(J)$. Indeed,*

$$\begin{aligned} U_D(I) &= min\{1 - I(a), I(a)\} + min\{1 - I(b), I(b)\} \\ &= 0.1 + 0.1 = 0.2, \qquad \textit{and} \\ U_D(J) &= 0.4 + 0.4 = 0.8. \end{aligned}$$

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

Observe that the program Π in Example 4 satisfies $MUFAS(\mathscr{L}_\infty, \Pi, U_D) = FAS(\mathscr{L}_2, \Pi)$, that is, its minimal undefined $\mathscr{L}_\infty$-answer sets coincide with its classical answer sets. This is not by chance, as claimed in the next theorem.

Theorem 3 (Congruence). *If Π is an $\mathscr{L}_2$-coherent program, then $MUFAS(\mathscr{L},\Pi,U) = FAS(\mathscr{L}_2,\Pi)$ for all sets of truth degrees $\mathscr{L}$ and measures of undefinedness U.*

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

Notice that if Π is $\mathscr{L}_2$-incoherent, then it may be the case that $MUFAS(\mathscr{L},\Pi,U) \neq FAS(\mathscr{L}_2,\Pi) = \emptyset$. For instance, the program $\Pi = \{a \leftarrow {\sim}a\}$ has no classical answer sets, but $\{(a,0.5)\}$ is in $MUFAS(\mathscr{L}_\infty,\Pi,U_D)$.

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 Π be $\{a \leftarrow {\sim}b \otimes c,\ b \leftarrow {\sim}a \otimes c,\ c \leftarrow {\sim}c\}$. For every $x \in [0,0.5]$, $I_x := \{(a,0.5-x),(b,x),(c,0.5)\}$ is an $\mathscr{L}_\infty$-answer set of Π with $U_D(I_x) = 1$. Moreover, there is no $J \in FAS(\mathscr{L}_\infty,\Pi)$ with $U_D(J) < 1$. Hence, Π has uncountably many minimal undefined fuzzy answer sets.* ■

Computation

Anytime algorithms (Alviano, Dodaro, and Ricca 2014; Alviano and Dodaro 2016) for computing minimal undefinedness fuzzy answer sets are now presented. The underlying idea is to compute one fuzzy answer set, and iteratively search for new fuzzy answer sets of lower undefinedness. As shown in (Alviano and Peñaloza 2015), fuzzy answer sets can be computed via a rewriting into satisfiability modulo theories (SMT): given an $\mathscr{L}$-program Π, the computed fuzzy answer set I is represented by the assignment to real constants x_p, for all atoms occurring $At(\Pi)$; formally, for all $p \in At(\Pi)$, it holds that $x_p = I(p)$.

The rewritings presented by Alviano and Peñaloza, whose details are not relevant for the scope of this paper, can be extended to compute fuzzy answer sets with a measure of undefinedness bounded by a given real number. For example, if the measure of undefinedness is U_D, and one is interested to fuzzy answer sets whose undefinedness is at most a given bound $b \in \mathbb{R}$, any rewriting from (Alviano and Peñaloza 2015) can be extended with the formula

$$\sum_{p \in At(\Pi)} ite(x_p < 1 - x_p, x_p, 1 - x_p) \leq b \qquad (6)$$

where $ite(x_p < 1 - x_p, x_p, 1 - x_p)$ essentially evaluates to the minimum between x_p and $1 - x_p$ (see (Alviano and Peñaloza 2015) for a formal definition).

In the following, for a measure function U, and a bound $b \in \mathbb{R}$, we will use the formula

$$U(\{(p,x_p) \mid p \in At(\Pi)\}) \leq b \qquad (7)$$

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

Algorithm 1: BinarySearch(Π, U, ε)

```
1 $(sat, I^*) := solve(\Pi)$;
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     $(sat, I) := solve(\Pi, U(\{(p,x_p) \mid p \in At(\Pi)\}) \leq b)$;
7     if $sat$ then $(I^*, ub) := (I, U(I))$;
8     else $lb := b$;
9 return $I^*$;
```

The first algorithm we consider is based on *binary search* (Algorithm 1). Its input is an $\mathscr{L}$-program, a measure of undefinedness U, and a precision threshold $\varepsilon \in \mathbb{R}^+$. The algorithm returns either the string *incoherent*, if $FAS(\mathscr{L},\Pi) = \emptyset$ (lines 1–2), or an $\mathscr{L}$-answer set $I^* \in FAS(\mathscr{L},\Pi)$ such that $U(I^*) - U(J) < \varepsilon$ for all $J \in MUFAS(\mathscr{L},\Pi,U)$. If Π 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 $\mathscr{L}$-answer set of Π. Then, the algorithm searches for an $\mathscr{L}$-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 $I \in FAS(\mathscr{L},\Pi)$ 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 ε.

Example 6. *Consider the program $\Pi = \{a \leftarrow {\sim}b, b \leftarrow {\sim}a\}$ from Example 4, with the measure U_D, and $\varepsilon = 0.1$. $solve(\Pi)$ returns a fuzzy answer set of Π; 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 ε (line 3), doubled at each iteration (line 6), and reset to ε 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 $I_1 = \{(a,0.3),$*

Algorithm 2: ProgressionSearch(Π, U, ε)

```
$(sat, I^*) := solve(\Pi)$;
**if not** $sat$ **then return** $incoherent$;
$(lb, ub, pr) := (-\varepsilon, U(I^*), \varepsilon)$;
**while** $ub - lb > \varepsilon$ **do**
    **if** $ub - pr \leq lb$ **then** $pr := \varepsilon$;
    $(b, pr) := (ub - pr, 2 \cdot pr)$;
    $(sat, I) := solve(\Pi, U(\{(p, x_p) \mid p \in At(\Pi)\}) \leq b)$;
    **if** $sat$ **then** $(I^*, ub) := (I, U(I))$;
    **else** $lb := b$;
**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* $U_D(I_3) \leq 0.1$ *is seeked. 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 ε*-improvement*. This method is obtained from Algorithm 1 by replacing line 5 to update b as follows:

```
5   $b := ub - \varepsilon$
```

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

Example 8. *Consider the input from Example 6, yielding the first solution* $I_0 = \{(a, 0.6), (b, 0.4)\}$ *and* $lb \leftarrow 0, up \leftarrow 0.8$*. The next solution should have a measure of at most* 0.7*; hence, the algorithm retrieves e.g.* $I_1 = \{(a, 0.3), (b, 0.7)\}$*. At the next iteration, b is set to* 0.5*, which yields a new solution like* $I_2 = \{(a, 0.1), (b, 0.9)\}$*. The next 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.* ■

Implementation and Evaluation

The three algorithms presented above were implemented in a prototype system extending FASP2SMT (Alviano and Peñaloza 2015) with the distance function U_D 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 $\mathscr{L}$-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 $\mathscr{L}$-program Π, which is parsed and rewritten into SMT-Lib (Barrett, Stump, and Tinelli 2010), and processed by Z3 (de Moura and Bjørner 2008) to compute an $\mathscr{L}$-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 (Mushthofa, Schockaert, and Cock 2014; Alviano and Peñaloza 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 ε is set to 0.1, and anyhow the majority of the testcases if ε is smaller. The value of ε significantly affects the algorithm based on progression. Indeed, this algorithm reached the best performance when ε was set to 0.01. The larger value 0.1 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 ε-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 ε-improvement.

Related Work

FAS and *MUFAS* are related to paracoherent answer set semantics. First we consider 3-*valued stable models* semantics (one of the best known approximations of answer sets) introduced in (Przymusinski 1991). Each 3-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-valued stable models of the program Π from Example 4 are $\{a, {\sim}b\}$, $\{{\sim}a, b\}$, and $\emptyset$, which correspond to $\{(a, 1), (b, 0)\}$, $\{(a, 0), (b, 1)\}$ and $\{(a, 0.5), (b, 0.5)\}$, respectively. On the other hand, Π has infinitely many fuzzy answer sets. However, if we restrict to the $\mathscr{L}_3$ semantics for programs using only the

Table 1: Experimental results on Hamiltonian Cycle instances: solved instances and average bound difference after 600 seconds; average running time and memory consumption on solved instances is also reported.

ε		binary	progress	ε-impr.	`min.`
0.1	solved	100%	22%	0%	6%
	$ub - lb$	0.073	0.409	7.157	–
	time (s)	149	206	–	10
	mem. (MB)	60	59	–	68
0.01	solved	94%	67%	6%	0%
	$ub - lb$	0.013	0.111	7.834	–
	time (s)	227	295	419	0
	mem. (MB)	61	62	74	0
0.001	solved	83%	44%	0%	0%
	$ub - lb$	0.005	0.565	8.399	–
	time (s)	230	358	–	0
	mem. (MB)	61	62	–	0

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 $\mathscr{L}_3$. Finally, among other paracoherent answer set semantics, we consider *Semi-stable models* (Sakama and Inoue 1995) and *Semi-equilibrium models* (Amendola et al. 2016). These paracoherent semantics satisfy three desiderata (see (Amendola et al. 2016)): (i) every consistent answer set of a program corresponds to a paracoherent model (*answer set coverage*); (ii) if a program has some (consistent) answer set, then its paracoherent models correspond to answer sets (*congruence*); (iii) if a program has a classical model, then it has a paracoherent model (*classical coherence*). In general, *FAS* semantics (and, thus, *MUFAS* semantics) does not satisfy this last property. For example, the program $\Pi = \{0 \leftarrow {\sim}a \mathbin{\bar{\wedge}} {\sim}b \mathbin{\bar{\wedge}} {\sim}c,\ a \leftarrow {\sim}b,\ b \leftarrow {\sim}c,\ c \leftarrow {\sim}a\}$ 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 (Hu et al. 2010; Vagin and Fomina 2011; Wang 2011; Wang and Dong 2009; 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 paracoherent 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 abduction 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 $-\varepsilon$. For example, for $\Pi = \{a \leftarrow 0.1 \mathbin{\underline{\vee}} {\sim}b,\ b \leftarrow 0.8 \mathbin{\bar{\wedge}} {\sim}a\}$ and $I \in FAS(\mathscr{L}, \Pi)$, $I(a) \in [0.1, 0.2]$ and $I(b) \in [0.8, 0.9]$ hold. Hence, *lb* in Algorithms 1–3 can be safely initialized to $0.2 - \varepsilon$.

References

Alviano, M., and Dodaro, C. 2016. Anytime answer set optimization via unsatisfiable core shrinking. *TPLP* 16(5-6):533–551.

Alviano, M., and Peñaloza, R. 2013. Fuzzy answer sets approximations. *TPLP* 13(4-5):753–767.

Alviano, M., and Peñaloza, R. 2015. Fuzzy answer set computation via satisfiability modulo theories. *TPLP* 15(4-5):588–603.

Alviano, M.; Calimeri, F.; Charwat, G.; and et al. 2013. The fourth answer set programming competition: Preliminary report. In Cabalar, P., and Son, T. C., eds., *LPNMR 2013*, volume 8148 of *LNCS*, 42–53.

Alviano, M.; Dodaro, C.; and Ricca, F. 2014. Anytime computation of cautious consequences in answer set programming. *TPLP* 14(4-5):755–770.

Amendola, G.; Eiter, T.; Fink, M.; Leone, N.; and Moura, J. 2016. Semi-equilibrium models for paracoherent answer set programs. *Artif. Intell.* 234:219–271.

Barrett, C.; Stump, A.; and Tinelli, C. 2010. The SMT-LIB Standard: Version 2.0. In Gupta, A., and Kroening, D., eds., *SMT'10*.

Bhandari, D., and Pal, N. R. 1993. Some new information measures for fuzzy sets. *Inf. Sci.* 67(3):209–228.

Blondeel, M.; Schockaert, S.; Vermeir, D.; and Cock, M. D. 2013. Fuzzy answer set programming: An introduction. In *Soft Computing: State of the Art Theory and Novel Applications*. 209–222.

Cintula, P.; Hájek, P.; and Noguera, C., eds. 2011. *Handbook of Mathematical Fuzzy Logic*, volume 37–38 of *Studies in Logic*. College Publications.

De Luca, A., and Termini, S. 1972. A definition of a non-probabilistic entropy in the setting of fuzzy sets theory. *Information and Control* 20(4):301–312.

De Luca, A., and Termini, S. 1974. Entropy of L-fuzzy sets. *Information and Control* 24(1):55–73.

de Moura, L. M., and Bjørner, N. 2008. Z3: an efficient SMT solver. In *TACAS 2008*, 337–340.

Eiter, T.; Leone, N.; and Saccà, D. 1997. On the partial semantics for disjunctive deductive databases. *Ann. Math. Artif. Intell.* 19(1-2):59–96.

Gebser, M.; Kaminski, R.; König, A.; and Schaub, T. 2011. Advances in *gringo* series 3. In *LPNMR 2011*, 345–351.

Gelfond, M., and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Comput.* 9(3/4):365–386.

Haber, R. E.; del Toro, R. M.; and Gajate, A. 2010. Optimal fuzzy control system using the cross-entropy method. A case study of a drilling process. *Inf. Sci.* 180(14):2777–2792.

Hu, Q.; Pan, W.; An, S.; Ma, P.; and Wei, J. 2010. An efficient gene selection technique for cancer recognition based on neighborhood mutual information. *Int. J. Machine Learning & Cybernetics* 1(1-4):63–74.

Janssen, J.; Schockaert, S.; Vermeir, D.; and Cock, M. D. 2012a. *Answer Set Programming for Continuous Domains - A Fuzzy Logic Approach*, volume 5 of *Atlantis Computational Intelligence Systems*. Atlantis Press.

Janssen, J.; Vermeir, D.; Schockaert, S.; and Cock, M. D. 2012b. Reducing fuzzy answer set programming to model finding in fuzzy logics. *TPLP* 12(6):811–842.

Jayne, E. 1957. Information theory and statistical mechanics. *Physical Reviews* 106(4):620–630.

Kapur, J. N., and Kesavan, H. K. 1992. Entropy optimization principles and their applications. In *Entropy and Energy Dissipation in Water Resources*. Springer Netherlands. 3–20.

Kosko, B. 1986. Fuzzy entropy and conditioning. *Inf. Sci.* 40(2):165–174.

Kullback, S. 1959. *Information Theory and Statistics*. New York: Wiley.

Lee, J., and Wang, Y. 2014. Stable models of fuzzy propositional formulas. In *JELIA 2014*, 326–339.

Li, X., and Liu, B. 2007. Maximum entropy principle for fuzzy variables. *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems* 15(Supplement-2):43–52.

Li, X. 2015. Fuzzy cross-entropy. *J. of Uncertainty Analysis and Applications* 3(1):1–6.

Lukasiewicz, T. 2006. Fuzzy description logic programs under the answer set semantics for the semantic web. In *RuleML 2006*, 89–96. IEEE Computer Society.

Marek, V. W., and Truszczyński, M. 1999. Stable Models and an Alternative Logic Programming Paradigm. In *The Logic Programming Paradigm – A 25-Year Perspective*. Springer Verlag. 375–398.

Mushthofa, M.; Schockaert, S.; and Cock, M. D. 2014. A finite-valued solver for disjunctive fuzzy answer set programs. In *ECAI 2014*, 645–650.

Mushthofa, M.; Schockaert, S.; and Cock, M. D. 2015. Solving disjunctive fuzzy answer set programs. In *LPNMR 2015*, 453–466.

Niemelä, I. 1999. Logic programs with stable model semantics as a constraint programming paradigm. *Ann. Math. Artif. Intell.* 25(3-4):241–273.

Nieuwenborgh, D. V.; Cock, M. D.; and Vermeir, D. 2007. An introduction to fuzzy answer set programming. *Ann. Math. Artif. Intell.* 50(3-4):363–388.

Pal, N., and Pal, S. 1992. Higher order fuzzy entropy and hybrid entropy of a set. *Inf. Sci.* 61(3):211–231.

Parkash, O.; Sharma, P.; and Mahajan, R. 2008. New measures of weighted fuzzy entropy and their applications for the study of maximum weighted fuzzy entropy principle. *Inf. Sci.* 178(11):2389–2395.

Przymusinski, T. C. 1991. Stable semantics for disjunctive programs. *New Generation Comput.* 9(3/4):401–424.

Qin, Z.; Li, X.; and Ji, X. 2009. Portfolio selection based on fuzzy cross-entropy. *J. Comput. App. Math.* 228(1):139–149.

Sakama, C., and Inoue, K. 1995. Paraconsistent stable semantics for extended disjunctive programs. *J. Log. Comput.* 5(3):265–285.

Shang, X., and Jiang, W. 1997. A note on fuzzy information measures. *Pattern Recognition Letters* 18(5):425–432.

Szmidt, E., and Kacprzyk, J. 2001. Entropy for intuitionistic fuzzy sets. *Fuzzy Sets and Systems* 118(3):467–477.

Vagin, V. N., and Fomina, M. V. 2011. Problem of knowledge discovery in noisy databases. *Int. J. Machine Learning & Cybernetics* 2(3):135–145.

Wang, X., and Dong, C. 2009. Improving generalization of fuzzy IF-THEN rules by maximizing fuzzy entropy. *IEEE Trans. Fuzzy Systems* 17(3):556–567.

Wang, X.; Dong, L.; and Yan, J. 2012. Maximum ambiguity-based sample selection in fuzzy decision tree induction. *IEEE Trans. Knowl. Data Eng.* 24(8):1491–1505.

Wang, X.; Zhai, J.; and Lu, S. 2008. Induction of multiple fuzzy decision trees based on rough set technique. *Inf. Sci.* 178(16):3188–3202.

Wang, L. 2011. An improved multiple fuzzy NNC system based on mutual information and fuzzy integral. *Int. J. Machine Learning & Cybernetics* 2(1):25–36.

Xie, W., and Bedrosian, S. 1984. An information measure for fuzzy sets. *IEEE Transactions on systems, man, and cybernetics* 14(1):151–156.

Xie, H.; Zheng, Y.; Guo, J.; and Chen, X. 2010. Cross-fuzzy entropy: A new method to test pattern synchrony of bivariate time series. *Inf. Sci.* 180(9):1715–1724.

Yager, R. R. 1979. On the measure of fuzziness and negation part i: Membership in the unit interval. *Int. J. of General Systems* 5(4):221–229.

Yager, R. R. 1980. On the measure of fuzziness and negation. II. Lattices. *Information and Control* 44(3):236–260.

Yi, W.; Lu, M.; and Liu, Z. 2011. Multi-valued attribute and multi-labeled data decision tree algorithm. *Int. J. Machine Learning & Cybernetics* 2(2):67–74.

You, J., and Yuan, L. 1994. A three-valued semantics for deductive databases and logic programs. *J. Comput. Syst. Sci.* 49(2):334–361.

Zhai, J. 2011. Fuzzy decision tree based on fuzzy-rough technique. *Soft Comput.* 15(6):1087–1096.