Fuzzy Answer Sets Approximations

Mario Alviano\textsuperscript{1} and Rafael Peñaloza\textsuperscript{2}

\textsuperscript{1}Department of Mathematics and Computer Science
University of Calabria

\textsuperscript{2}Center for Advancing Electronics Dresden
Dresden University of Technology

ICLP 2013
Istanbul, 27 August 2013
Outline

1 Introduction

2 Syntax and Semantics

3 Approximation Operators
   - Immediate Consequence Operator
   - Minimal Satisfiability
   - Well-founded Operator

4 Implementation and Experiment

5 Conclusion
Answer Set Programming (ASP)

- Overcomes a weakness of classical logic for KR: monotonicity
- Naturally handles reasoning by defaults, abductive reasoning, belief revisions, ...

ASP makes logic closer to the real world

However...

- Everything is either true or false in ASP
- ASP is based on precise information

Can we always make these assumptions?
Answer Set Programming (ASP)

- Overcomes a weakness of classical logic for KR: monotonicity
- Naturally handles reasoning by defaults, abductive reasoning, belief revisions, ...

ASP makes logic closer to the real world

However...

- Everything is either true or false in ASP
- ASP is based on precise information

Can we always make these assumptions?
Barber of Seville paradox

In the small town of Seville, all and only those men who do not shave themselves are shaved by the barber (who is a man). Who shaves the barber?

\[\text{shaves(barber, } X) \leftarrow \text{not shaves}(X, X)\]
\[\text{shaves}(X, X) \leftarrow \text{not shaves(barber, } X)\]

- Classical set theory can neither prove nor disprove that the barber shaves himself
- An odd loop makes the program incoherent for ASP
- \text{shaves(barber, barber)} is undefined according to the well-founded semantics
Barber of Seville paradox

In the small town of Seville, all and only those men who do not shave themselves are shaved by the barber (who is a man). Who shaves the barber?

Classical set theory can neither prove nor disprove that the barber shaves himself.

An odd loop makes the program incoherent for ASP.

$shaves(barber, barber) \leftrightarrow \neg shaves(barber, barber)$

$shaves(barber, barber) \leftrightarrow \neg shaves(barber, barber)$

$shaves(barber, barber)$ is undefined according to the well-founded semantics.
Is everything true or false?

Let’s associate false statements with white, and true statements with black.

Undefined statements can be associated to gray.

Can we model statements which are more likely to be false or to be true?

... and we can go on!
Is everything true or false?

How many truth degrees?

- Let’s associate false statements with white, and true statements with black
- Undefined statements can be associated to gray
- Can we model statements which are more likely to be false or to be true?
- ... and we can go on!
Is everything true or false?

How many truth degrees?

- Let’s associate false statements with white, and true statements with black
- Undefined statements can be associated to gray
- Can we model statements which are more likely to be false or to be true?
- ... and we can go on!
Is everything true or false?

How many truth degrees?

- Let’s associate false statements with white, and true statements with black.
- Undefined statements can be associated to gray.
- Can we model statements which are more likely to be false or to be true?
- ... and we can go on!
Is everything true or false?

How many truth degrees?

- Let’s associate false statements with white, and true statements with black.
- Undefined statements can be associated to gray.
- Can we model statements which are more likely to be false or to be true?
- ... and we can go on!
Only precise information?

- When a person stops to be young and becomes old?

- Compare it with the classical, crisp approach
Fuzzy logic interprets propositions with a truth degree in $[0,1]$.

Classical (crisp) set vs. fuzzy set

Fuzzification

Fuzzy inference engine

Defuzzification

This work deals with the engine part.
Possible Application

Possible solution:

\[
\begin{align*}
\text{top}(X, \text{high}) & \leftarrow \text{top}(X, \text{medium}), \quad \text{not bottom}(X - 1, \text{high}). \\
\text{bottom}(X, \text{high}) & \leftarrow \text{bottom}(X, \text{medium}), \quad \text{not top}(X + 1, \text{high}). \\
\text{sign}(X, I) & \leftarrow \text{top}(X, I). \\
\text{sign}(X, I) & \leftarrow \text{bottom}(X, I).
\end{align*}
\]
Possible solution

$$\text{top}(X, \text{high}) \leftarrow \text{top}(X, \text{medium}),$$
$$\text{not bottom}(X - 1, \text{high}).$$

$$\text{bottom}(X, \text{high}) \leftarrow \text{bottom}(X, \text{medium}),$$
$$\text{not top}(X + 1, \text{high}).$$

$$\text{sign}(X, I) \leftarrow \text{top}(X, I).$$
$$\text{sign}(X, I) \leftarrow \text{bottom}(X, I).$$
Fuzzy inference engine

- Minimal satisfiability operator
- Fuzzy unfounded sets
- Implementation
  - Grounder (gringo)
  - Solver: fuzzy answer sets (approximation) and well-founded model
- Experiment
Normal FASP program: set of rules of the form
\[ a ← b_1 ⊗ ⋯ ⊗ b_m ⊗ not b_{m+1} ⊗ ⋯ ⊗ not b_n \]

Fuzzy atoms are either propositional atoms in a fixed set \( B \) or numeric constants: \( a, b, speed, 0, 1, 0.5, 1/3, \ldots \)

Example
\[
\begin{align*}
a &← not b \\
b &← a ⊗ 0.8
\end{align*}
\]

\( ⊗ : [0, 1] × [0, 1] → [0, 1] \) is a fixed t-norm, e.g.

- Gödel: \( x ⊗ y = \min\{x, y\} \)
- Product: \( x ⊗ y = x \cdot y \)
- Łukasiewicz: \( x ⊗ y = \max\{x + y - 1, 0\} \)
Interpretation: \( I: \mathcal{B} \rightarrow [0, 1] \)

- Numeric constant: \( I(\overline{c}) = c \)
- Negative literal: \( I(\text{not } a) = 1 - I(a) \)
- Conjunction: \( I(\bigotimes_{i=1}^{k} l_i) = \bigotimes_{i=1}^{k} I(l_i) \)

Model: \( I \models r \) if \( I(H(r)) \geq I(B(r)) \)

Program Reduct: in \( P \), replace each \( \text{not } b \) by \( I(\text{not } b) \)

Fuzzy Answer Set: \( M \models P \) and there is no \( I \subset M \) s.t. \( I \models P^M \)

---

**Fuzzy Sets: Operations and Relations**

- \( I \subseteq J \): \( I(a) \leq J(a) \) for each \( a \in \mathcal{B} \)
- \( I \subset J \): \( I \subseteq J \) and \( I \neq J \)
- \( I \cap J \): \( a \mapsto \min\{I(a), J(a)\} \) for each \( a \in \mathcal{B} \)
- \( I \cup J \): \( a \mapsto \max\{I(a), J(a)\} \) for each \( a \in \mathcal{B} \)
- \( I \setminus J \): \( a \mapsto \max\{I(a) - J(a), 0\} \) for each \( a \in \mathcal{B} \)
Outline

1. Introduction

2. Syntax and Semantics

3. Approximation Operators
   - Immediate Consequence Operator
   - Minimal Satisfiability
   - Well-founded Operator

4. Implementation and Experiment

5. Conclusion
Goal: Compute Approximations

- Start with the largest approximation \((0, 1)\)
  - Lower bound: all propositions associated with 0
  - Upper bound: all propositions associated with 1
- Apply operators to restrict the approximation
  - Increase the lower bound
  - Decrease the upper bound
- All the answer sets must be between these bounds
Immediate Consequence Operator

**Definition**

For each \( a \in B \), define

\[
T_{P}^{U}(L) : a \mapsto \max\{\langle L, U \rangle (r) \mid r \in P, H(r) = a\}
\]

For \( r : a \leftarrow b_{1} \otimes \cdots \otimes b_{m} \otimes \neg b_{m+1} \otimes \cdots \otimes \neg b_{n} \)

let \( \langle L, U \rangle (r) := L(b_{1} \otimes \cdots \otimes b_{m}) \otimes U(\neg b_{m+1} \otimes \cdots \otimes \neg b_{n}) \)

**Łukasiewicz t-norm:** \( x \otimes y = \max\{x + y - 1, 0\} \)

\[
a \leftarrow b \otimes \neg c
\]

\[
L = \{ a \mapsto 0, b \mapsto 0.8, c \mapsto 0.3 \}
\]

\[
U = \{ a \mapsto 1, b \mapsto 1, c \mapsto 0.3 \}
\]

\[
T_{P}^{U}(L) : a \mapsto L(b) \otimes U(\neg c) = \max\{0.8 + (1 - 0.3) - 1, 0\} = 0.5
\]

**Theorem**

The fixpoint \( T_{P}^{U} \uparrow 0 \) is reached after a linear number of iterations, measured on the number of atoms appearing in \( P \).
Immediate Consequence Operator

**Definition**

For each \( a \in \mathcal{B} \), define

\[
T^U_P(L) : a \mapsto \max \{ \langle L, U \rangle (r) \mid r \in P, H(r) = a \}
\]

For \( r : a \leftarrow b_1 \otimes \cdots \otimes b_m \otimes \neg b_{m+1} \otimes \cdots \otimes \neg b_n \)

let \( \langle L, U \rangle (r) := L(b_1 \otimes \cdots \otimes b_m) \otimes U(\neg b_{m+1} \otimes \cdots \otimes \neg b_n) \)

**Łukasiewicz t-norm:** \( x \otimes y = \max\{x + y - 1, 0\} \)

\[
a \leftarrow b \otimes \neg c \quad L = \{ a \mapsto 0, b \mapsto 0.8, c \mapsto 0.3 \}
\]

\[
U = \{ a \mapsto 1, b \mapsto 1, c \mapsto 0.3 \}
\]

\[
T^U_P(L) : a \leftarrow L(b) \otimes U(\neg c) = \max\{0.8 + (1 - 0.3) - 1, 0\} = 0.5
\]

**Theorem**

The fixpoint \( T^U_P \uparrow 0 \) is reached after a linear number of iterations, measured on the number of atoms appearing in \( P \).
Definition

For each \( a \in B \), define

\[
S^U_P(L): a \mapsto \inf\{I(a) \mid I \models P \land L \subseteq I \subseteq U\}
\]

Theorem

If \( M \models P \) and \( L \subseteq M \subseteq U \) then \( S^U_P(L) \subseteq M \subseteq U \).

How to compute this operator?

For the Łukasiewicz t-norm, rewrite each

\[
a \leftarrow b_1 \otimes \cdots \otimes b_m \otimes \text{not } b_{m+1} \otimes \cdots \otimes \text{not } b_n
\]

into

\[
a \geq b_1 + \cdots + b_m - b_{m+1} - \cdots - b_n + 1 - m
\]

and solve a system of linear inequalities.
Definition
For each $a \in \mathcal{B}$, define
\[ S^U_P(L) : a \mapsto \inf\{ I(a) \mid I \models P \land L \subseteq I \subseteq U \} \]

Theorem
If $M \models P$ and $L \subseteq M \subseteq U$ then $S^U_P(L) \subseteq M \subseteq U$.

How to compute this operator?
For the Łukasiewicz t-norm, rewrite each
\[ a \leftarrow b_1 \otimes \cdots \otimes b_m \otimes \text{not } b_{m+1} \otimes \cdots \otimes \text{not } b_n \]
into
\[ a \geq b_1 + \cdots + b_m - b_{m+1} - \cdots - b_n + 1 - m \]
and solve a system of linear inequalities.
### Example

<table>
<thead>
<tr>
<th>Rule</th>
<th>Minimize</th>
<th>Constraints</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a \leftarrow \neg b$</td>
<td>$\min a$</td>
<td>$a \geq 1 - b$</td>
<td>$S_P^1(0) : a \mapsto 0.6$</td>
</tr>
<tr>
<td>$b \leftarrow a \otimes 0.8$</td>
<td>$\min b$</td>
<td>$b \geq a + 0.8 - 1$</td>
<td>$S_P^1(0) : b \mapsto 0.4$</td>
</tr>
</tbody>
</table>

### Theorem

The minimal satisfiability operator is computable in polynomial time for programs over the Łukasiewicz t-norm.
Fuzzy Unfounded Sets

Definition

A fuzzy unfounded set \( X \) for \( P \) w.r.t. \((L, U)\) satisfies

\[
[U \cap (1 \setminus X)](H(r)) \geq \langle U \cap (1 \setminus X), L \rangle (r)
\]

for each \( r \in P \) such that \( X(H(r)) > 0 \).

Fuzzy unfounded sets evidence lack of (acyclic) support.

Łukasiewicz t-norm: \( x \otimes y = \max\{x + y - 1, 0\} \)

\[
\begin{align*}
\text{Unfounded sets for } (0, 1): \\
a \leftarrow b \otimes b \\
b \leftarrow a \\
b \leftarrow 0.9 \otimes \text{not } c \\
c \leftarrow 0.9 \otimes \text{not } b
\end{align*}
\]

\[
\begin{align*}
X_1 &= \{a \mapsto 0, b \mapsto 0, c \mapsto 0\} \\
X_2 &= \{a \mapsto 0, b \mapsto 0, c \mapsto 0.1\} \\
X_3 &= \{a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0\} \\
X_4 &= \{a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0.1\}
\end{align*}
\]
A fuzzy unfounded set $X$ for $P$ w.r.t. $(L, U)$ satisfies 
$$[U \cap (1 \setminus X)](H(r)) \geq \langle U \cap (1 \setminus X), L \rangle (r)$$
for each $r \in P$ such that $X(H(r)) > 0$.

Fuzzy unfounded sets evidence lack of (acyclic) support.

Łukasiewicz t-norm: $x \otimes y = \max\{x + y - 1, 0\}$

<table>
<thead>
<tr>
<th>Unfounded sets for $(0, 1)$:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a \leftarrow b \otimes b$</td>
</tr>
<tr>
<td>$b \leftarrow a$</td>
</tr>
<tr>
<td>$b \leftarrow 0.9 \otimes \text{not } c$</td>
</tr>
<tr>
<td>$c \leftarrow 0.9 \otimes \text{not } b$</td>
</tr>
</tbody>
</table>

$X_1 = \{ a \mapsto 0, b \mapsto 0, c \mapsto 0 \}$

$X_2 = \{ a \mapsto 0, b \mapsto 0, c \mapsto 0.1 \}$

$X_3 = \{ a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0 \}$

$X_4 = \{ a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0.1 \}$
Fuzzy Unfounded Sets

Definition

A fuzzy unfounded set $X$ for $P$ w.r.t. $(L, U)$ satisfies

$$[U \cap (1 \setminus X)](H(r)) \geq \langle U \cap (1 \setminus X), L \rangle (r)$$

for each $r \in P$ such that $X(H(r)) > 0$.

Fuzzy unfounded sets evidence lack of (acyclic) support.

Łukasiewicz t-norm: $x \otimes y = \max\{x + y - 1, 0\}$

Unfounded sets for $(0, 1)$:

- $a \leftarrow b \otimes b$
- $b \leftarrow a$
- $b \leftarrow 0.9 \otimes \text{not } c$
- $c \leftarrow 0.9 \otimes \text{not } b$

$X_1 = \{a \mapsto 0, b \mapsto 0, c \mapsto 0\}$

$X_2 = \{a \mapsto 0, b \mapsto 0, c \mapsto 0.1\}$

$X_3 = \{a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0\}$

$X_4 = \{a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0.1\}$
Definition

A fuzzy unfounded set $X$ for $P$ w.r.t. $(L, U)$ satisfies
\[ [U \cap (1 \setminus X)](H(r)) \geq \langle U \cap (1 \setminus X), L \rangle (r) \]
for each $r \in P$ such that $X(H(r)) > 0$.

Fuzzy unfounded sets evidence lack of (acyclic) support.

Łukasiewicz t-norm: $x \otimes y = \max\{x + y - 1, 0\}$

Unfounded sets for $(0, 1)$:

- $a \leftarrow b \otimes b$
- $b \leftarrow a$
- $b \leftarrow 0.9 \otimes \neg c$
- $c \leftarrow 0.9 \otimes \neg b$

$X_1 = \{ a \mapsto 0, b \mapsto 0, c \mapsto 0 \}$
$X_2 = \{ a \mapsto 0, b \mapsto 0, c \mapsto 0.1 \}$
$X_3 = \{ a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0 \}$
$X_4 = \{ a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0.1 \}$
A fuzzy unfounded set $X$ for $P$ w.r.t. $(L, U)$ satisfies 

$$\left[ U \cap (1 \setminus X) \right](H(r)) \geq \langle U \cap (1 \setminus X), L \rangle (r)$$

for each $r \in P$ such that $X(H(r)) > 0$.

Fuzzy unfounded sets evidence lack of (acyclic) support.

Łukasiewicz t-norm: $x \otimes y = \max\{x + y - 1, 0\}$

Unfounded sets for $(0, 1)$:

- $a \leftarrow b \otimes b$
- $b \leftarrow a$
- $b \leftarrow 0.9 \otimes \text{not } c$
- $c \leftarrow 0.9 \otimes \text{not } b$

$X_1 = \{ a \mapsto 0, b \mapsto 0, c \mapsto 0 \}$

$X_2 = \{ a \mapsto 0, b \mapsto 0, c \mapsto 0.1 \}$

$X_3 = \{ a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0 \}$

$X_4 = \{ a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0.1 \}$
Lemma

The union of two (fuzzy) unfounded sets is an unfounded set.

Let $GUS^L_P, U_P$ denote the greatest fuzzy unfounded set.

Theorem

$M$ is a fuzzy answer set of a program $P$ iff $GUS^M_P, M = 1 \setminus M$.

Theorem

For programs without numeric constants and crisp sets, fuzzy unfounded sets are the unfounded sets by Van Gelder et al. [1991].
**Lemma**

The union of two (fuzzy) unfounded sets is an unfounded set.

Let $GUS^L_U$ denote the greatest fuzzy unfounded set.

**Theorem**

$M$ is a fuzzy answer set of a program $P$ iff $GUS^M_P = 1 \setminus M$.

**Theorem**

For programs without numeric constants and crisp sets, fuzzy unfounded sets are the unfounded sets by Van Gelder et al. [1991].
Well-founded Operator

\[ W_P(L, U) := (T_P^U(L), 1 \setminus R_P^{L;U} \downarrow 1) \]

where for every \( a \in \mathcal{B} \), define

\[ R_P^{L;U}(X) : a \mapsto \min\{X(a), 1 - \max\{\langle U \cap (1 \setminus X), L \rangle(r) \mid r \in P, H(r) = a\}\} \]

**Theorem**

The fixpoint of \( W_P \) gives the well-founded semantics by Damásio and Pereira [2001].
Implementation

FASP program → Grounder gringo → Numeric format

fasp

UpperBoundIncrease → UpperBoundDecrease → LowerBoundIncrease

MinimalSatisfiability

Bilevel solver YALMIP+Octave

Fuzzy answer set or approximation

https://github.com/alviano/fasp.git
Example

\[ a \leftarrow \text{not } b \quad \text{min } a - a' + b - b' \]
\[ b \leftarrow a \otimes 0.8 \quad \text{s.t. } 0 \leq a, b \leq 1 \]
\[ \text{min } a + b \]
\[ \text{s.t. } a' \geq 1 - b \]
\[ b' \geq a' + 0.8 - 1 \]
\[ 0 \leq a', b' \leq 1 \]

Fuzzy answer set or approximation

https://github.com/alviano/fasp.git
<table>
<thead>
<tr>
<th></th>
<th>Tested instances</th>
<th>Timeouts Unopt.</th>
<th>Opt.</th>
<th>Average execution time* Unopt.</th>
<th>Optimized</th>
<th>Average perc. gain*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph Coloring</td>
<td>60</td>
<td>34</td>
<td>0</td>
<td>247.44</td>
<td>34.45 (2.68)</td>
<td>76.43%</td>
</tr>
<tr>
<td>Hamiltonian Path</td>
<td>40</td>
<td>33</td>
<td>9</td>
<td>120.51</td>
<td>6.41 (0.02)</td>
<td>81.49%</td>
</tr>
<tr>
<td>Stratified</td>
<td>90</td>
<td>10</td>
<td>0</td>
<td>190.07</td>
<td>1.80 (0.02)</td>
<td>96.71%</td>
</tr>
<tr>
<td>Odd Cycle</td>
<td>90</td>
<td>33</td>
<td>0</td>
<td>186.94</td>
<td>1.95 (0.03)</td>
<td>97.18%</td>
</tr>
</tbody>
</table>

* Computed on the instances solved by both the approaches.
Outline

1. Introduction

2. Syntax and Semantics

3. Approximation Operators
   - Immediate Consequence Operator
   - Minimal Satisfiability
   - Well-founded Operator

4. Implementation and Experiment

5. Conclusion
Conclusion

- Approximations provide a sensitive performance gain
- Still the bilevel approach seems to be too slow

Future Work

- More complex structures in the body (disjunction)
  - Compile into uninterpreted function symbols in gringo
  - Use a pre-processor in fasp
- How to deal with a choice operator?
  - Completion approach by Janssen et al. [2012]
  - Conflict analysis and learning?

Thank you!
Conclusion

- Approximations provide a sensitive performance gain
- Still the bilevel approach seems to be too slow

Future Work

- More complex structures in the body (disjunction)
  - Compile into uninterpreted function symbols in gringo
  - Use a pre-processor in fasp
- How to deal with a choice operator?
  - Completion approach by Janssen et al. [2012]
  - Conflict analysis and learning?

Thank you!
