Fuzzy Answer Sets Approximations

Mario Alviano[1] and Rafael Peñaloza[2]

[1] Department of Mathematics and Computer Science
University of Calabria

[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 revisons, ...

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 revisons, ...

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?

$$shaves(barber, X) \leftarrow not\ shaves(X, X)$$
$$shaves(X, X) \leftarrow not\ shaves(barber, X)$$

- Classical set theory can neither prove nor disprove that the barber shaves himself
- An odd loop makes the program incoherent for ASP
- $shaves(barber, barber)$ 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?

$$shaves(barber, barber) \leftarrow not\ shaves(barber, barber)$$
$$shaves(barber, barber) \leftarrow not\ shaves(barber, barber)$$

- Classical set theory can neither prove nor disprove that the barber shaves himself
- An odd loop makes the program incoherent for ASP
- $shaves(barber, barber)$ is undefined according to the well-founded semantics

A B

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!

A D C E B

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!

A F D G C H E I B

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? (2)

A F D G C H E I B

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? (2)

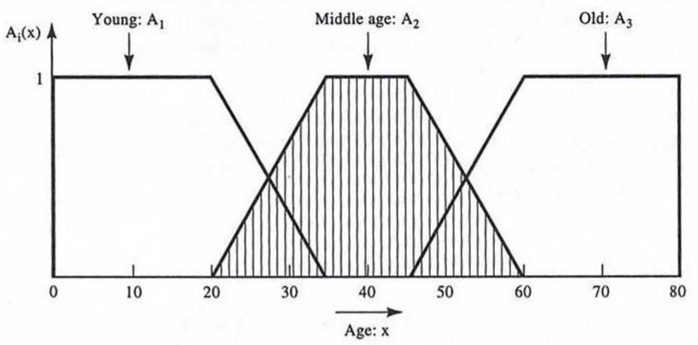
A F D G C H E I B

How many truth degrees?

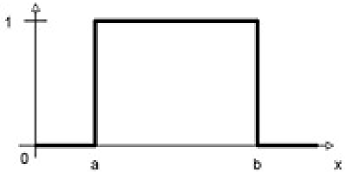
- Let's associate false statements with white, and true statements with black
- Undefined statements can be associated to gray
- Can we model statements which are more likely to be false or to be true?
- ... and we can go on!

Only precise information?

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



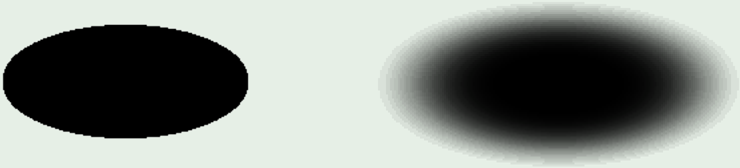
- Compare it with the classical, crisp approach

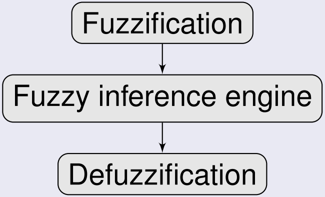


How to Handle Vagueness?

Fuzzy logic interprets propositions with a truth degree in [0,1]

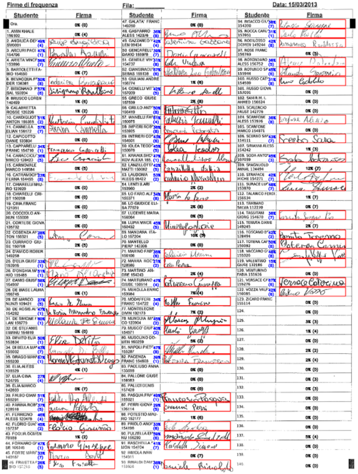
Classical (crisp) set vs. fuzzy set

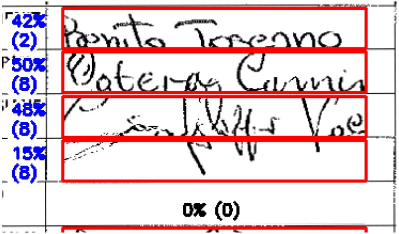




This work deals with the engine part

Possible Application

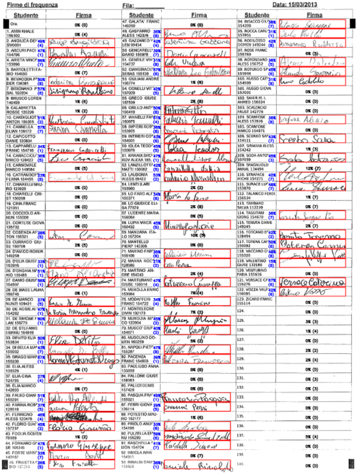


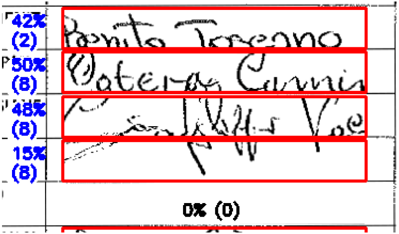


Possible solution

$$top(X, high) \leftarrow top(X, medium),$$
$$\qquad not\ bottom(X-1, high).$$
$$bottom(X, high) \leftarrow bottom(X, medium),$$
$$\qquad not\ top(X+1, high).$$
$$sign(X, I) \leftarrow top(X, I).$$
$$sign(X, I) \leftarrow bottom(X, I).$$

Possible Application





Possible solution

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

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

$$sign(X, I) \leftarrow top(X, I).$$

$$sign(X, I) \leftarrow bottom(X, I).$$

Fuzzy inference engine

- Minimal satisfiability operator
- Fuzzy unfounded sets
- Implementation
  - Grounder (gringo)
  - Solver: fuzzy answer sets (approximation) and well-founded model
- Experiment

Outline

1. Introduction
2. Syntax and Semantics
3. Approximation Operators
   - Immediate Consequence Operator
   - Minimal Satisfiability
   - Well-founded Operator
4. Implementation and Experiment
5. Conclusion

Normal FASP program: set of rules of the form

$$a \leftarrow b_1 \otimes \cdots \otimes b_m \otimes \mathit{not}\ b_{m+1} \otimes \cdots \otimes \mathit{not}\ b_n$$

Fuzzy atoms are either propositional atoms in a fixed set $\mathcal{B}$ or numeric constants: a, b, $speed$, $\overline{0}$, $\overline{1}$, $\overline{0.5}$, $\overline{1/3}$, ...

Example

$$a \leftarrow \mathit{not}\ b$$
$$b \leftarrow a \otimes \overline{0.8}$$

$\otimes : [0,1] \times [0,1] \rightarrow [0,1]$ is a fixed **t-norm**, e.g.

- Gödel: $x \otimes y = \min\{x, y\}$
- Product: $x \otimes y = x \cdot y$
- Łukasiewicz: $x \otimes y = \max\{x + y - 1, 0\}$

Interpretation: $I : \mathcal{B} \rightarrow [0,1]$

- Numeric constant: $I(\overline{c}) = c$
- Negative literal: $I(not\ a) = 1 - I(a)$
- 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 $not\ b$ by $\overline{I(not\ b)}$

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 $(\mathbf{0}, \mathbf{1})$
  - 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 $a \in \mathcal{B}$, define

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

For r : $a \leftarrow b_1 \otimes \cdots \otimes b_m \otimes not\ b_{m+1} \otimes \cdots \otimes not\ b_n$

let $\langle L, U\rangle(r) := L(b_1 \otimes \cdots \otimes b_m) \otimes U(not\ b_{m+1} \otimes \cdots \otimes not\ b_n)$

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

$a \leftarrow b \otimes not\ c$

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

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

$T_P^U(L) : a \mapsto L(b) \otimes U(not\ c) = \max\{0.8 + (1 - 0.3) - 1, 0\} = 0.5$

Theorem

The fixpoint $T_P^U \Uparrow \mathbf{0}$ 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

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

For $r:\ a \leftarrow b_1 \otimes \cdots \otimes b_m \otimes \mathit{not}\ b_{m+1} \otimes \cdots \otimes \mathit{not}\ b_n$

let $\langle L, U\rangle\,(r) := L(b_1 \otimes \cdots \otimes b_m) \otimes U(\mathit{not}\ b_{m+1} \otimes \cdots \otimes \mathit{not}\ b_n)$

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

$a \leftarrow b \otimes \mathit{not}\ c$

$$L = \{a \mapsto 0\ \ , b \mapsto 0.8, c \mapsto 0.3\}$$
$$U = \{a \mapsto 1\ \ , b \mapsto 1\ \ , c \mapsto 0.3\}$$

$$T_P^U(L) : a \mapsto L(b) \otimes U(\mathit{not}\ c) = \max\{0.8 + (1 - 0.3) - 1, 0\} = 0.5$$

Theorem

The fixpoint $T_P^U\Uparrow\mathbf{0}$ is reached after a linear number of iterations, measured on the number of atoms appearing in P.

Definition

For each $a \in \mathcal{B}$, define

$S_P^U(L) : a \mapsto \inf\{I(a) \mid I \models P \land L \subseteq I \subseteq U\}$

Theorem

If $M \models P$ and $L \subseteq M \subseteq U$ then $S_P^U(L) \subseteq M \subseteq U$.

How to compute this operator?

For the Łukasiewicz t-norm, rewrite each

$a \leftarrow b_1 \otimes \cdots \otimes b_m \otimes \textit{not } b_{m+1} \otimes \cdots \otimes \textit{not } b_n$

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
$$S_P^U(L) : a \mapsto \inf\{I(a) \mid I \models P \wedge L \subseteq I \subseteq U\}$$

Theorem

If $M \models P$ and $L \subseteq M \subseteq U$ then $S_P^U(L) \subseteq M \subseteq U$.

How to compute this operator?

For the Łukasiewicz t-norm, rewrite each
$$a \leftarrow b_1 \otimes \cdots \otimes b_m \otimes \textit{not } b_{m+1} \otimes \cdots \otimes \textit{not } b_n$$
into
$$a \geq b_1 + \cdots + b_m - b_{m+1} - \cdots - b_n + 1 - m$$
and solve a system of linear inequalities.

Example

$a \leftarrow \textit{not}\ b$

$b \leftarrow a \otimes \overline{0.8}$

$$\begin{aligned} \min\ & a \\ \text{s.t.}\ & a \geq 1 - b \\ & b \geq a + 0.8 - 1 \\ & 0 \leq a, b \leq 1 \end{aligned}$$

$S^{\mathbf{1}}_{P}(\mathbf{0}) : a \mapsto 0.6$

$$\begin{aligned} \min\ & b \\ \text{s.t.}\ & a \geq 1 - b \\ & b \geq a + 0.8 - 1 \\ & 0 \leq a, b \leq 1 \end{aligned}$$

$S^{\mathbf{1}}_{P}(\mathbf{0}) : b \mapsto 0.4$

Theorem

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

Definition

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

Fuzzy unfounded sets evidence lack of (acyclic) support.

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

$a \leftarrow b \otimes b$
$b \leftarrow a$
$b \leftarrow \overline{0.9} \otimes \mathit{not}\ c$
$c \leftarrow \overline{0.9} \otimes \mathit{not}\ b$

Unfounded sets for $(\mathbf{0}, \mathbf{1})$:

$X_1 = \{a \mapsto 0, b \mapsto 0, c \mapsto 0\}$
$X_2 = \{a \mapsto 0, b \mapsto 0, c \mapsto 0.1\}$
$X_3 = \{a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0\}$
$X_4 = \{a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0.1\}$

Definition

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

Fuzzy unfounded sets evidence lack of (acyclic) support.

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

$a \leftarrow b \otimes b$
$b \leftarrow a$
$b \leftarrow \overline{0.9} \otimes \mathit{not}\ c$
$c \leftarrow \overline{0.9} \otimes \mathit{not}\ b$

Unfounded sets for $(\mathbf{0}, \mathbf{1})$:

$X_1 = \{a \mapsto 0, b \mapsto 0, c \mapsto 0\}$
$X_2 = \{a \mapsto 0, b \mapsto 0, c \mapsto 0.1\}$
$X_3 = \{a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0\}$
$X_4 = \{a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0.1\}$

Definition

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

$$[U \cap (\mathbf{1} \setminus X)](H(r)) >= \langle U \cap (\mathbf{1} \setminus X), L \rangle (r)$$

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

Fuzzy unfounded sets evidence lack of (acyclic) support.

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

$a \leftarrow b \otimes b$
$b \leftarrow a$
$b \leftarrow \overline{0.9} \otimes not\ c$
$c \leftarrow \overline{0.9} \otimes not\ b$

Unfounded sets for $(\mathbf{0}, \mathbf{1})$:

$X_1 = \{a \mapsto 0, b \mapsto 0, c \mapsto 0\}$
$X_2 = \{a \mapsto 0, b \mapsto 0, c \mapsto 0.1\}$
$X_3 = \{a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0\}$
$X_4 = \{a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0.1\}$

Definition

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

$$[U \cap (\mathbf{1} \setminus X)](H(r)) >= \langle U \cap (\mathbf{1} \setminus X), L \rangle (r)$$

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

Fuzzy unfounded sets evidence lack of (acyclic) support.

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

$a \leftarrow b \otimes b$
$b \leftarrow a$
$b \leftarrow \overline{0.9} \otimes \mathit{not}\ c$
$c \leftarrow \overline{0.9} \otimes \mathit{not}\ b$

Unfounded sets for $(\mathbf{0}, \mathbf{1})$:

$X_1 = \{a \mapsto 0, b \mapsto 0, c \mapsto 0\}$
$X_2 = \{a \mapsto 0, b \mapsto 0, c \mapsto 0.1\}$
$X_3 = \{a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0\}$
$X_4 = \{a \mapsto 0.2, b \mapsto 0.1, c \mapsto 0.1\}$

Definition

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

Fuzzy unfounded sets evidence lack of (acyclic) support.

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

$a \leftarrow b \otimes b$

$b \leftarrow a$

$b \leftarrow \overline{0.9} \otimes \textit{not } c$

$c \leftarrow \overline{0.9} \otimes \textit{not } b$

Unfounded sets for $(\mathbf{0}, \mathbf{1})$:

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

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

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

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

Fuzzy Unfounded Sets (2)

Lemma

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

Let $GUS_P^{L,U}$ denote the greatest fuzzy unfounded set.

Theorem

M is a fuzzy answer set of a program P iff $GUS_P^{M,M} = \mathbf{1} \setminus M$.

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 $GUS_P^{L,U}$ denote the greatest fuzzy unfounded set.

Theorem

M is a fuzzy answer set of a program P iff $GUS_P^{M,M} = \mathbf{1} \setminus M$.

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

$$W_P(L, U) := \left(T_P^U(L), \mathbf{1} \setminus R_P^{L,U}\Downarrow\mathbf{1}\right)$$

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

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

Theorem

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

Outline

1 Introduction

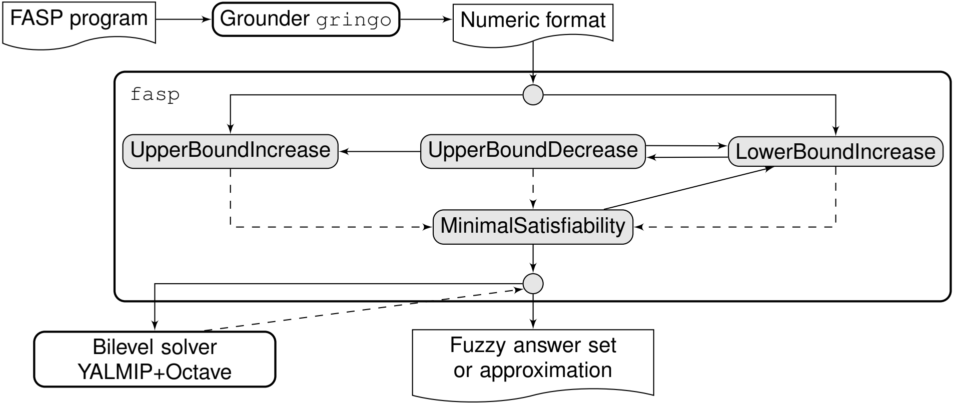
2 Syntax and Semantics

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

4 Implementation and Experiment

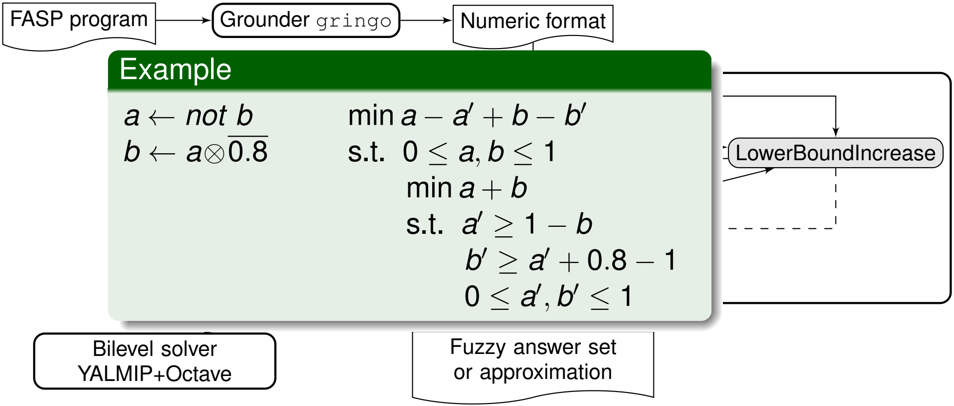
5 Conclusion

Implementation



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

Implementation



Example

$a \leftarrow not\ b$

$b \leftarrow a \otimes \overline{0.8}$

$$\min a - a' + b - b'$$
$$\text{s.t. } 0 \le a, b \le 1$$
$$\min a + b$$
$$\text{s.t. } a' \ge 1 - b$$
$$b' \ge a' + 0.8 - 1$$
$$0 \le a', b' \le 1$$

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

Experiment

	Tested instances	Timeouts		Average execution time*		Average perc. gain*
		Unopt.	Opt.	Unopt.	Optimized	
Graph Coloring	60	34	0	247.44	34.45 (2.68)	76.43%
Hamiltonian Path	40	33	9	120.51	6.41 (0.02)	81.49%
Stratified	90	10	0	190.07	1.80 (0.02)	96.71%
Odd Cycle	90	33	0	186.94	1.95 (0.03)	97.18%

* 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!

Bibliography

Carlos Viegas Damásio and Luís Moniz Pereira. Antitonic logic programs. In Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning, LPNMR'01, pages 379–392, London, UK, UK, 2001. Springer-Verlag. ISBN 3-540-42593-4.

Jeroen Janssen, Dirk Vermeir, Steven Schockaert, and Martine De Cock. Reducing fuzzy answer set programming to model finding in fuzzy logics. Theory and Practice of Logic Programming, 12(6):811–842, 2012.

Allen Van Gelder, Kenneth A. Ross, and John S. Schlipf. The Well-Founded Semantics for General Logic Programs. Journal of the ACM, 38(3):620–650, 1991.