Approximation Fixpoint Theory and its Application to Knowledge Representation

Bart Bogaerts

Department of computer science, Vrije Universiteit Brussel

January 15, 2019

Approximation Fixpoint Theory

Given

- *Complete lattice* $\langle L, \leq \rangle$
- *(Approximation) bilattice* $\langle L^2, \leq, \leq_p \rangle$
- *Lattice operator* $O : L \to L$,
- *Approximating bilattice operator* $A : L^2 \to L^2$:
  - $O(x) \in A(x, x)$
  - $\leq_p$ *-monotone*

Often assumed

Symmetric: $A(x, y) = (A(y, x)_2, A(y, x)_1)$
Exact: $A(x, x) = (O(x), O(x))$

Stable operator

$S_A(x, y) \mapsto (\text{lfp}\, A(\cdot, y)_1, \text{lfp}\, A(x, \cdot)_2)$

Fixpoints

- *Supported:* $O(x) = x$
- *Partial supported* $A(x, y) = (x, y)$
- *Partial stable* $S_A(x, y) = (x, y)$
- *Stable:* x *s.t.* (x, x) *is partial stable*
- *Kripke-Kleene:* $\text{lfp}_{\leq_p} A$
- *Well-founded:* $\text{lfp}_{\leq_p} S_A$
- *Grounded:* x *s.t.* $\forall v : O(x \wedge v) \leq v \Rightarrow x \leq v$.

Content

1. History & Origin
2. Further theoretical advances
3. Application Domains
   - Weighted Abstract Dialectic Frameworks
4. Future work
5. Conclusion

Logic Programming

Predicate Logic as a Programming Language, van Emden and Kowalski, 1976

- Rules of the form

$$
\begin{aligned}
& r(start). \\
& r(X) \leftarrow e(Y, X) \wedge r(Y). \\
& nr(X) \leftarrow node(X) \wedge \neg r(X).
\end{aligned}
$$

- Declarative semantics?
- Meaning of negation (as failure)?

Clark's completion semantics

Negation as Failure, Clark, 1978

logic program = *definition*

Completion semantics (supported models):

$$\forall X : r(X) \Leftrightarrow (X = start) \vee (\exists Y : e(Y, X) \wedge r(Y)).$$
$$\forall X : nr(X) \Leftrightarrow node(X) \wedge \neg r(X).$$

Problem with recursion: self-supporting arguments

Minimal model semantics

Semantics only considers minimal models of logic program (viewed as a set of implications).
Works well (avoids self-supporting arguments)... for positive programs.

Perfect model semantics

On the Declarative Semantics of Deductive Databases and Logic Programs, Przymusinski, 1988

- Stratification over atoms/symbols.
- Only refer negatively to symbols from lower strata.
- At each stratum: minimal model semantics.
- ... Only works for (locally) stratified programs.

Intermezzo

In this talk: programs are assumed to be ground (no first-order variables)
Can always be satisfied by taking (possibly infinite) grounding

$$r(start).$$
$$r(a) \leftarrow e(start, a) \wedge r(start).$$
$$r(b) \leftarrow e(start, b) \wedge r(start).$$
$$r(c) \leftarrow e(start, c) \wedge r(start).$$
$$\ldots$$
$$r(a) \leftarrow e(a, a) \wedge r(a).$$
$$r(a) \leftarrow e(b, a) \wedge r(b).$$
$$r(a) \leftarrow e(e, a) \wedge r(e).$$
$$\ldots$$
$$nr(start) \leftarrow node(start) \wedge \neg r(start).$$
$$nr(a) \leftarrow node(a) \wedge \neg r(a).$$

$A \wedge B$		B: t	f	u
A	t	t	f	u
	f	f	f	f
	u	u	f	u

$A \vee B$		B: t	f	u
A	t	t	t	t
	f	t	f	u
	u	t	u	u

		$\neg A$
A	t	f
	f	t
	u	u

Figure: The Kleene truth tables.

Standard three-valued truth evaluation φ^I

Figure: The truth order $\leq$ and the precision order $\leq_p$

Note: $\wedge$ is $\text{glb}_{\leq}$ and $\vee$ is $\text{lub}_{\leq}$

$A \wedge B$		B: **t**	**f**	**u**	**i**
A	**t**	**t**	**f**	**u**	**i**
	f	**f**	**f**	**f**	**f**
	u	**u**	**f**	**u**	**f**
	i	**i**	**f**	**f**	**i**

$A \vee B$		B: **t**	**f**	**u**	**i**
A	**t**	**t**	**t**	**t**	**t**
	f	**t**	**f**	**u**	**i**
	u	**t**	**u**	**u**	**t**
	i	**t**	**i**	**t**	**i**

		$\neg A$
A	**t**	**f**
	f	**t**
	u	**u**
	i	**i**

Figure: The Kleene truth tables.

Standard four-valued truth evaluation φ^I

Figure: The truth order $\leq$ and the precision order $\leq_p$

Note: $\wedge$ is $\text{glb}_{\leq}$ and $\vee$ is $\text{lub}_{\leq}$

Well-founded semantics

The Well-Founded Semantics for General Logic Programs, Van Gelder, Ross and Schlipf, 1991 (1988)

Defined through sequence (I_n) of three-valued interpretations (each atom is assigned $\mathbf{t}, \mathbf{f}$, or $\mathbf{u}$). Two types of refinements:

- given I_i, update value of p to $\text{lub}_{\leq}\{body(r)^{I_i} \mid head(r) = p\}$
- unfounded set: set $I_{i+1}(p) = \mathbf{f}$ for each p in a set P of atoms such that after doing this, they stay false (for each rule r with $head(r) \in P$: $body(r)^{I_{i+1}} = \mathbf{f}$).

Stable semantics

The Stable Model Semantics for Logic Programming, Gelfond and Lifschitz, 1988

Reduct of $\mathcal{P}$ with respect to M:

$$\{p \leftarrow a_1 \wedge \cdots \wedge a_n \mid (p \leftarrow a_1 \ldots \ldots a_n \wedge \neg b_1 \wedge \cdots \wedge \neg b_m) \in \mathcal{P} \wedge \forall i : b_i^M = \mathbf{f}\}$$

Definition

M is a stable model if it is the minimal model of $\mathcal{P}^M$

In the meanwhile...

Non-monotonic reasoning: a.o.,[1]

Semantical considerations on nonmonotonic logic, Moore, 1985 (autoepistemic logic)

A Logic for Default Reasoning, Reiter, 1980 (default logic)

Similar problems with self-supporting arguments

[1] Many more non-monotonic logics were founded back then!

Example: Autoepistemic Logic

- Classical logic + epistemic operator K ("I know")
- "All I know" assumption

Example

$$p.$$

$$Kp \Rightarrow q.$$

$$K\neg q \Rightarrow r.$$

Example: Autoepistemic Logic

- Classical logic + epistemic operator K ("I know")
- "All I know" assumption

Example

$$p.$$
$$Kp \Rightarrow q.$$
$$K\neg q \Rightarrow r.$$

Example

$$Kp \Rightarrow p.$$

Back to logic programming

- Fixpoint semantics
- Idea: associate an operator with a logic program
- Define its semantics in terms of fixpoints of that operator

Fixpoint semantics for logic programming — A survey, Fitting, 2002

A semantic operator

Definition

Let $\mathcal{P}$ be a logic program. The operator $T_{\mathcal{P}}$ maps an interpretation I to

$$T_{\mathcal{P}}(I) = \{p \mid \exists r \in \mathcal{P} : head(r) = p \wedge I \models body(r)\}.$$

- Intuitively: update I according to $\mathcal{P}$.
- Already defined by van Emden and Kowalski
- Fixpoints of $T_{\mathcal{P}}$ are supported models
- What about stable/minimal/well-founded/... models? Fitting also characterized them.

Approximation Fixpoint Theory

- Generalization of Fitting's theory:
  - Within logic programming: using semantic instead of syntactic constructions (simpler operator)
  - Applicable to any operator on a complete lattice (various non-monotonic domains)
- Same types of fixpoints
- Identifies the underlying semantic principles
- Now... what are they?

Stable Operators, Well-Founded Fixpoints and Applications in Nonmonotonic Reasoning, Denecker, Marek and Truszczyński, 2000

A three/four- valued semantic operator (an approximator)

Definition

Let $\mathcal{P}$ be a logic program. The operator $\Psi_{\mathcal{P}}$ maps a partial interpretation I to

$$\Psi_{\mathcal{P}}(I) : p \mapsto \text{lub}_{\leq}\{body(r)^{I} \mid r \in \mathcal{P} \wedge head(r) = p\}.$$

Some observations

- $\Psi_{\mathcal{P}}$ coincides with $T_{\mathcal{P}}$ on two-valued interpretations
- $\Psi_{\mathcal{P}}$ is $\leq_p$-monotone
- Preserves consistency

Intermezzo: tuple-representation

- Represent a four-valued truth value as two two-valued truth values

$$\mathbf{t} = (\mathbf{t}, \mathbf{t}) \quad \mathbf{f} = (\mathbf{f}, \mathbf{f}) \quad \mathbf{u} = (\mathbf{f}, \mathbf{t}) \quad \mathbf{i} = (\mathbf{t}, \mathbf{f})$$

- First: lower bound; second: upper bound
- Alternative view: interval in $\mathbf{f} \leq \mathbf{t}$
- Alternative representation of four-valued interpretation: tuple (I_1, I_2) of two two-valued interpretations
- I_1 all certainly true atoms; I_2 all possibly true atoms

Stable fixpoints

- Idea of reduct: make assumption about false atoms, see if you can derive true atoms
- Rephrased in terms of approximator: I is stable iff

$$I = \operatorname{lfp} \Psi_{\mathcal{P}}(\cdot, I)$$

- $\Psi_{\mathcal{P}}(\cdot, I)$ is immediate consequence operator of reduct $\mathcal{P}^I$!

Approximation Fixpoint Theory

Given

- *Complete lattice* $\langle L, \leq \rangle$
- *(Approximation) bilattice* $\langle L^2, \leq, \leq_p \rangle$
- *Lattice operator* $O : L \to L$,
- *Approximating bilattice operator* $A : L^2 \to L^2$:
  - $O(x) \in A(x, x)$
  - $\leq_p$ *-monotone*

Often assumed

Symmetric: $A(x, y) = (A(y, x)_2, A(y, x)_1)$
Exact: $A(x, x) = (O(x), O(x))$

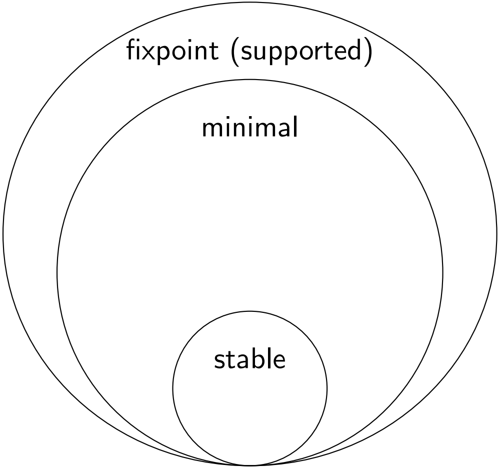
Stable operator

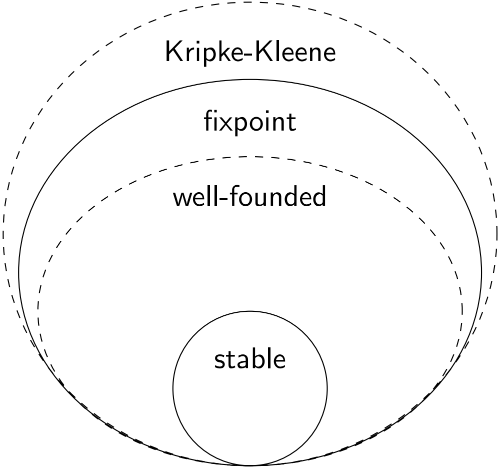
$S_A(x, y) \mapsto (\text{lfp}\, A(\cdot, y)_1, \text{lfp}\, A(x, \cdot)_2)$

Fixpoints

- *Supported:* $O(x) = x$
- *Partial supported* $A(x, y) = (x, y)$
- *Partial stable* $S_A(x, y) = (x, y)$
- *Stable:* x *s.t.* (x, x) *is partial stable*
- *Kripke-Kleene:* $\text{lfp}_{\leq_p} A$
- *Well-founded:* $\text{lfp}_{\leq_p} S_A$
- *Grounded:* x *s.t.* $\forall v : O(x \wedge v) \leq v \Rightarrow x \leq v$.

Relationships between fixpoints





well-founded = least precise partial stable

Kripke-Kleene = least precise partial supported

Why use AFT?

- Insight in underlying principles
- Unification of domains (sometimes surprising results)
- Provides confidence (well-established principles of non-monotonic reasoning)
- Can highlight issues in semantics
- Results are more general (applicable to all fields captured by AFT)
- Transfer of existing results

Example: DL and AEL

- Default logic (DL):

$$Bird(x) \wedge M\ Fly(x) \Rightarrow Fly(x)$$

- Autoepistemic logic (AEL):

$$Bird(x) \wedge \neg K \neg Fly(x) \Rightarrow Fly(x)$$

- Transformation from DL to AEL

On the relation between default and autoepistemic logic, Konolige, 1988

- Preserves intuitions, but not semantics
- Long-standing problem: relationship between DL and AEL?

Example: DL and AEL (AFT)

- Konolige's transformation preserves semantic operator
- Thus also all AFT-style semantics
- Reiter's DL semantics: stable fixpoints
- Moore's AEL semantics: supported fixpoints

Uniform semantic treatment of default and autoepistemic logics, Denecker, Marek and Truszczyński, 2003

Reiter's Default Logic Is a Logic of Autoepistemic Reasoning And a Good One, Too, Denecker, Marek and Truszczyński, 2011

Content

Ultimate approximator

- For most purposes it suffices to define approximators on consistent bilattice elements
- Consistent approximators can be ordered in a precision order
- Most precise approximator: ultimate approximator:

$$U_O : (x, y) \mapsto \mathrm{glb}_{\leq_p} \{O(z) \mid x \leq z \leq y\}$$

- Approximator derived from operator

Ultimate approximation and its application in nonmonotonic knowledge representation systems, Denecker, Marek and Truszczyński, 2004

Modularity

Determine fixpoints in terms of fixpoint of smaller operators

- Strong equivalence and uniform equivalence

Strong and uniform equivalence of nonmonotonic theories - an algebraic approach , Truszczyński, 2006

- Stratification

Splitting an operator: Algebraic modularity results for logics with fixpoint semantics, Vennekens et al, 2006

- Predicate introduction

Predicate Introduction for Logics With a Fixpoint Semantics, Vennekens et al, 2007

Constructions

- Constructive characterization of semantics
- E.g. for well-founded semantics

Definition

An *A-refinement* of (x, y) is a pair $(x', y') \in L^2$ satisfying one of:

- $(x, y) \leq_p (x', y') \leq_p A(x, y)$, or
- $x' = x$ and $A(x, y')_2 \leq y' \leq y$.

Definition

A *well-founded induction* of A is a transfinite sequence $(x_i, y_i)_{i \leq \beta}$ with:

- $(x_0, y_0) = (\bot, \top)$;
- (x_{i+1}, y_{i+1}) is an A-refinement of (x_i, y_i), for all $i < \beta$;
- $(x_\lambda, y_\lambda) = \mathrm{lub}_{\leq_p} \{(x_i, y_i) \mid i < \lambda\}$ for each limit ordinal $\lambda \leq \beta$.

Constructions

Well-Founded Semantics and the Algebraic Theory of Non-monotone Inductive Definitions, Denecker and Vennekens, 2007

Safe inductions and their applications in knowledge representation, Bogaerts et al, 2018

Groundedness

- Idea that models should be "grounded"/should not contain self-supporting arguments shows up in many domains

$$p \leftarrow p.$$

$$Kp \Rightarrow p.$$

- Idea: formalize this directly in terms of fixpoints of operators

Definition (Grounded)

We call $x \in L$ *grounded* for O if for each $v \in L$ such that $O(x \wedge v) \leq v$, it holds that $x \leq v$.

Grounded fixpoints and their applications in knowledge representation, Bogaerts et al, 2015

Grounded fixpoints: intuitively

- Intuition, if $L = 2^F$, $\leq = \subseteq$
- Then: $\bigwedge = \bigcap$ and $\bigvee = \bigcup$

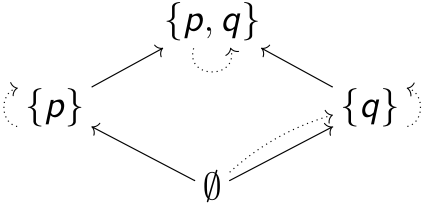
Definition (Grounded)

We call $x \in L$ *grounded* for O if for each $v \in L$ such that $O(x \wedge v) \leq v$, it holds that $x \leq v$.

x is grounded for O if it only contains facts that are sanctioned by O: whenever we remove facts from x, at least one of them is rederived.

Groundedness: Example

$$\left\{ \begin{array}{l} p \leftarrow p. \\ q \leftarrow \neg p \lor q. \end{array} \right\}$$



Approximation Fixpoint Theory

Given

- *Complete lattice* $\langle L, \leq \rangle$
- *(Approximation) bilattice* $\langle L^2, \leq, \leq_p \rangle$
- *Lattice operator* $O : L \to L$,
- *Approximating bilattice operator* $A : L^2 \to L^2$:
  - $O(x) \in A(x, x)$
  - $\leq_p$ *-monotone*

Often assumed

Symmetric: $A(x, y) = (A(y, x)_2, A(y, x)_1)$
Exact: $A(x, x) = (O(x), O(x))$

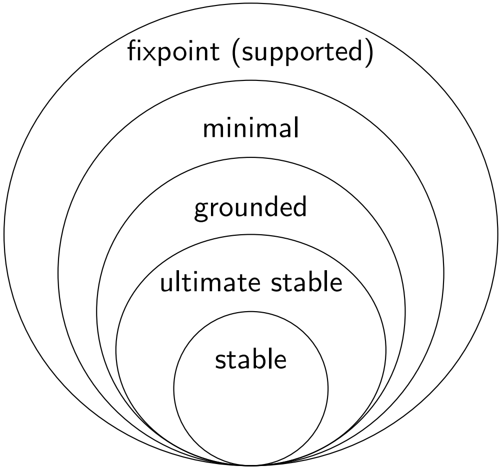
Stable operator

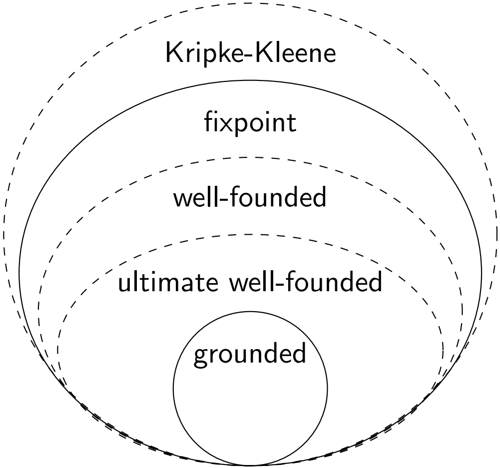
$S_A(x, y) \mapsto (\text{lfp}\, A(\cdot, y)_1, \text{lfp}\, A(x, \cdot)_2)$

Fixpoints

- *Supported:* $O(x) = x$
- *Partial supported* $A(x, y) = (x, y)$
- *Partial stable* $S_A(x, y) = (x, y)$
- *Stable:* x *s.t.* (x, x) *is partial stable*
- *Kripke-Kleene:* $\text{lfp}_{\leq_p} A$
- *Well-founded:* $\text{lfp}_{\leq_p} S_A$
- *Grounded:* x *s.t.* $\forall v : O(x \wedge v) \leq v \Rightarrow x \leq v$.

Relationships between fixpoints





well-founded = least precise partial stable = least precise partial grounded

Content

Extensions of logic programming

- Semantics for syntactic extensions
  - Aggregates in bodies
  - HEX atoms
  - Integration of description logics with logic programming
  - Higher-order logic programming

Aggregates

- Given a logic program with aggregates in the body, e.g.,

$$controls(C_1, C_2) \leftarrow \sum_{\{C_3,N|(controls(C_1,C_3)\vee C_1=C_3)\wedge OwnsShares(C_3,C_2,N)\}} N \geq 50$$

- How to extend stable/well-founded semantics preserving the underlying principles?
- Difficult problem (tens of papers)
- With AFT: easy to solve
  - Operator: straightforward
  - Ultimate approximator: directly obtained
  - Other approximators can be defined directly (similar to normal logic programs). All we need is a three-valued truth evaluation of aggregates

Well-founded and Stable Semantics of Logic Programs with Aggregates, Pelov et al, 2007

Active Integrity Constraints (AICs)

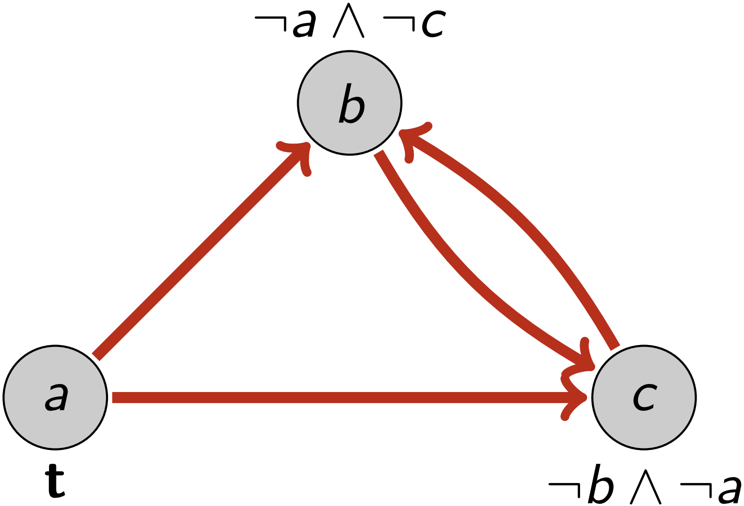
- Modern-day databases: *integrity constraints* essential
- What if such constraints are violated? How to fix the database?
- Active integrity contsraints: rules of the form

$$p \wedge q \wedge \neg r \supset -p$$

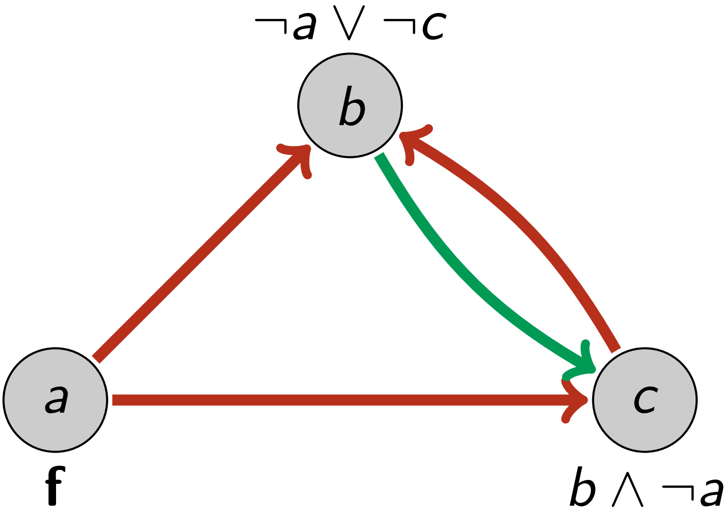
- Operator and approximator for AICs solve semantic problems in the field

Fixpoint Semantics for Active Integrity Constraints, Bogaerts and Cruz-Filipe, 2018

Dung's Argumentation Frameworks



Abstract Dialectical Frameworks (ADFs)

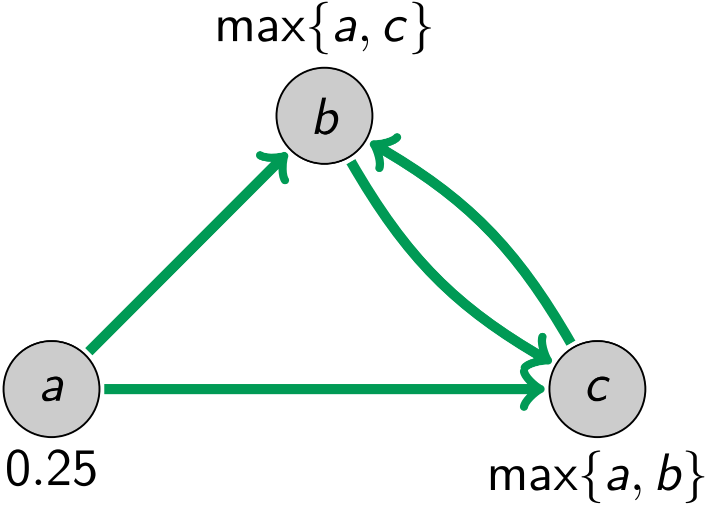


AFT for ADFs

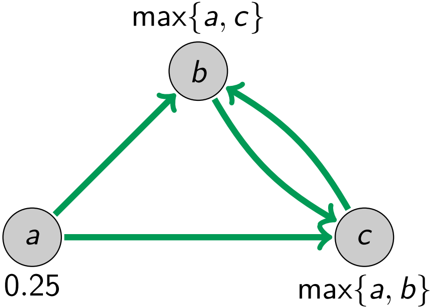
- Semantic operator and approximators for AFs and ADFs
- Recovered existing semantics and induced some new semantics
- Application of AFT required using *asymmetric* approximator, raised suspicion
- Ultimately, resulted in a revision of ADF semantics, using ultimate semantics

Approximating operators and semantics for abstract dialectical frameworks, Straß, 2013

Weighted Abstract Dialectical Frameworks



wADFs: Semantics (AAAI'18)



Over the unit interval

- *Models*: assignments I s.t. $I(a) = 0.25$ and $I(b) = I(c) \in [0.25, 1]$.
- *Grounded interpretation* assigns 0.25 to a and $\mathbf{u}$ to b and c.
- *W-stable models*: those models such that $I(b) \in W$

Weighted Abstract Dialectical Frameworks, Brewka et al, 2018

Open questions

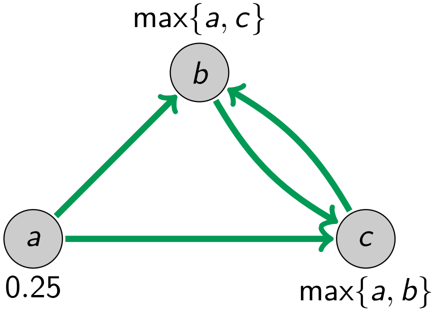
1. What are suitable **approximations** of interpretations?
2. How can we, systematically, generalize the asymmetry between true and false in ADFs to wADFs? Thus, how can we obtain a generalization of stable semantics (and of other semantics) in which smaller acceptance values are preferred over larger acceptance values?

AFT-style semantics for wADFs

- New formalization of wADFs based on Approximation Fixpoint Theory
- Solves both questions using **interval-based** approximations
- Identified issues in existing semantics

Weighted abstract dialectical frameworks through the lens of approximation fixpoint theory, Bogaerts, 2019

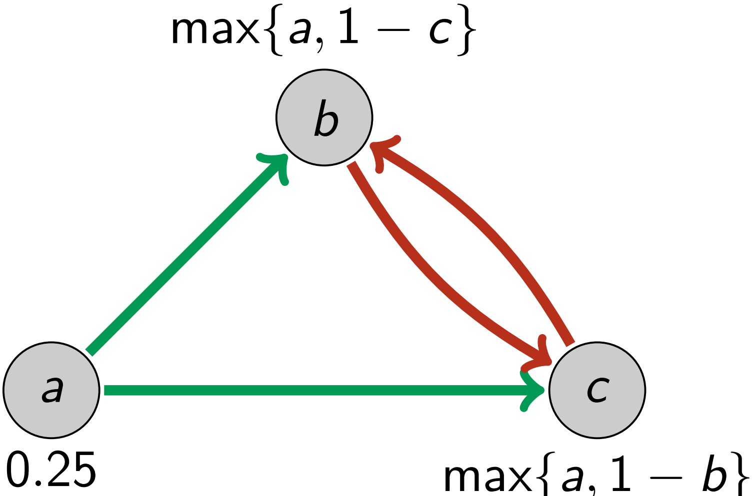
wADFs: AFT-style Semantics



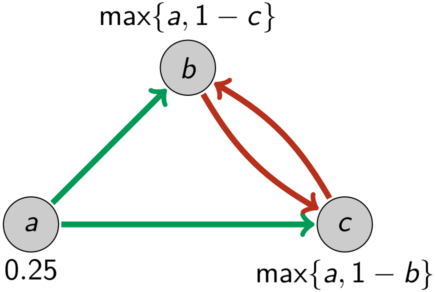
Over the unit interval

- *Models*: assignments I s.t. $I(a) = 0.25$ and $I(b) = I(c) \in [0.25, 1]$.
- *Grounded interpretation* assigns 0.25 to a and $[0.25, 1]$ to b and c,
- Unique *stable model* I: $I(a) = I(b) = I(c) = 0.25$.

Another Example



wADFs: AFT-style Semantics



Over the unit interval

- *(Stable) Models*: assignments I s.t. $I(a) = 0.25$ and $I(b) = 1 - I(c) \in [0.25, 0.75]$.
- *Grounded interpretation* assigns 0.25 to a and $[0.25, 0.75]$ to b and c,

Definitions

Definition

A *weighted abstract dialectical framework* over ν is a tuple $\Xi = (S, C)$, where

- S is a vocabulary, i.e. a set of arguments
- $C = \{C_s^{in}\}_{s \in S}$ is a collection of functions $C_s^{in} : int(\nu, S) \to \nu$.
- $\langle \nu, \leq_a \rangle$ forms a complete lattice

With a wADF, we associate an operator on partial interpretations:

Definition

$$U_\Xi(X) : s \mapsto \text{glb}_{\leq_p} \{C_s^{in}(Y) \mid Y \in int(\nu, S) \wedge Y \geq_p X\}$$

Definition

- The *grounded* ν^c*-interpretation* of Ξ is the least fixpoint of U^{ν}_{Ξ}
- A ν^c-interpretation (X, Y) is *admissible* with respect to Ξ if $(X, Y) \leq_p U^{\nu}_{\Xi}(X, Y)$.
- A ν^c-interpretation (X, Y) is *complete* with respect to Ξ if $(X, Y) = U^{\nu}_{\Xi}(X, Y)$.
- An interpretation X is a *model* of Ξ if (X, X) is complete with respect to Ξ.
- A partial interpretation (X, Y) is *stable* with respect to Ξ if it is a stable fixpoint of U^{ν}_{Ξ}.
- An interpretation X is a *stable model* of Ξ if (X, X) is stable.
- The *AFT-well-founded* ν^c*-interpretation* is the well-founded fixpoint of U^{ν}_{Ξ}.

Definition is a copy-paste of the ADF case. All we needed to do was define the operator!

Two Definitions of wADFs

ν-wADF $=$ a tuple (S, C) with:

- S a vocabulary
- C a family acceptance functions $C_s^{in} : int(\nu, S) \to \nu$.

BSWW (AAAI'18):

- ν is precision-ordered $(\leq_i)$
- Approximations in $\nu \cup \{\mathbf{u}\}$

AFT-style:

- ν is truth-ordered $(\leq_a)$
- Approximations: intervals in ν

Semantic operator defined (almost) identically!

wADFs: Conclusion

- Applied AFT to wADFs
  - Clarifies relationship with various NMR formalisms
  - New semantics (well-founded) and improved stable semantics
  - Semantics defined using well-established principles
  - Access to rich theory
- In-depth comparison of the two approaches
  - Identified troublesome design decision in wADF: lack of distincition between values and approximations
- Complexity analysis

Content

Justifications

- Justification theory: other unifying theory
- Focus on explainability (justification $\approx$ explanation)
- Induces an approximator
- Some overlap in application domains
- Relationship with AFT?

Domain Theory

- AFT explains *inductively defined sets* (captures many forms of mathematical induction)
- Domain theory: *recursively defined functions* and their *domain*
- Unify these two fields?

Non-determinism

- Extension of AFT to non-deterministic operators
- Disjunctive logic programming
- Causality

Approximation Spaces

- Fundamental assumption: approximations are intervals
- Several applications required modifications of this assumption
- What if we drop it?
- Generalization of towards arbitrary approximation spaces?

Content

Conclusion

- Framework for defining semantics of non-monotonic formalisms
- Rich theoretic foundation
- Many theoretic results obtained “for free” by applying AFT
- Provides insight in underlying principles
- Unification of domains (sometimes surprising results)
- Provides confidence
- Can highlight issues in semantics
- Applications have revealed several points with potential for improvement

Approximation Fixpoint Theory

Given

- *Complete lattice* $\langle L, \leq \rangle$
- *(Approximation) bilattice* $\langle L^2, \leq, \leq_p \rangle$
- *Lattice operator* $O : L \to L$,
- *Approximating bilattice operator* $A : L^2 \to L^2$:
  - $O(x) \in A(x, x)$
  - $\leq_p$ *-monotone*

Often assumed

Symmetric: $A(x, y) = (A(y, x)_2, A(y, x)_1)$
Exact: $A(x, x) = (O(x), O(x))$

Stable operator

$S_A(x, y) \mapsto (\text{lfp}\, A(\cdot, y)_1, \text{lfp}\, A(x, \cdot)_2)$

Fixpoints

- *Supported:* $O(x) = x$
- *Partial supported* $A(x, y) = (x, y)$
- *Partial stable* $S_A(x, y) = (x, y)$
- *Stable:* x *s.t.* (x, x) *is partial stable*
- *Kripke-Kleene:* $\text{lfp}_{\leq_p} A$
- *Well-founded:* $\text{lfp}_{\leq_p} S_A$
- *Grounded:* x *s.t.* $\forall v : O(x \wedge v) \leq v \Rightarrow x \leq v$.