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. \( \forall v : O(x \land v) \leq v \Rightarrow x \leq v \).
Logic Programming

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

- Rules of the form

  \[ r(\text{start}). \]
  \[ r(X) \leftarrow e(Y, X) \land r(Y). \]
  \[ nr(X) \leftarrow \text{node}(X) \land \neg r(X). \]

- Declarative semantics?
- Meaning of negation (as failure)?
Clark’s completion semantics

**Negation as Failure, Clark, 1978**

**logic program = definition**

Completion semantics (supported models):

\[
\forall X : r(X) \Leftrightarrow (X = \text{start}) \lor (\exists Y : e(Y, X) \land r(Y)).
\]

\[
\forall X : nr(X) \Leftrightarrow \text{node}(X) \land \neg r(X).
\]

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

\[ r(start). \]
\[ r(a) \leftarrow e(start,a) \land r(start). \]
\[ r(b) \leftarrow e(start,b) \land r(start). \]
\[ r(c) \leftarrow e(start,c) \land r(start). \]
\[ \ldots \]
\[ r(a) \leftarrow e(a,a) \land r(a). \]
\[ r(a) \leftarrow e(b,a) \land r(b). \]
\[ r(a) \leftarrow e(e,a) \land r(e). \]
\[ \ldots \]
\[ nr(start) \leftarrow node(start) \land \neg r(start). \]
\[ nr(a) \leftarrow node(a) \land \neg r(a). \]
Standard three-valued truth evaluation $\varphi'$

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

Note: $\land$ is $\text{glb}_{\leq}$ and $\lor$ is $\text{lub}_{\leq}$
### Standard four-valued truth evaluation $\varphi'$

<table>
<thead>
<tr>
<th>$A \land B$</th>
<th>$B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>$t$</td>
</tr>
<tr>
<td>$f$</td>
<td>$f$</td>
</tr>
<tr>
<td>$u$</td>
<td>$u$</td>
</tr>
<tr>
<td>$i$</td>
<td>$i$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$A \lor B$</th>
<th>$B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>$t$</td>
</tr>
<tr>
<td>$f$</td>
<td>$t$</td>
</tr>
<tr>
<td>$u$</td>
<td>$t$</td>
</tr>
<tr>
<td>$i$</td>
<td>$t$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$A$</th>
<th>$\neg A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>$f$</td>
</tr>
<tr>
<td>$f$</td>
<td>$t$</td>
</tr>
<tr>
<td>$u$</td>
<td>$u$</td>
</tr>
<tr>
<td>$i$</td>
<td>$i$</td>
</tr>
</tbody>
</table>

**Figure:** The Kleene truth tables.

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

Note: $\land$ is $\text{glb}_{\leq}$ and $\lor$ is $\text{lub}_{\leq}$
Well-founded semantics


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

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

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

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

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

Definition

$M$ is a stable model if it is the minimal model of $\mathcal{P}^M$
Non-monotonic reasoning: a.o.,\textsuperscript{1}

\textit{Semantical considerations on nonmonotonic logic, Moore, 1985 (autoepistemic logic)}

\textit{A Logic for Default Reasoning, Reiter, 1980 (default logic)}

Similar problems with self-supporting arguments

\textsuperscript{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

\[
\begin{align*}
p. \\
Kp \Rightarrow q. \\
K\neg q \Rightarrow r.
\end{align*}
\]
Example: Autoepistemic Logic

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

Example

\[
\begin{align*}
& p. \\
& Kp \Rightarrow q. \\
& K\neg q \Rightarrow r.
\end{align*}
\]

Example

\[
Kp \Rightarrow p.
\]
Fixpoint semantics
Idea: associate an operator with a logic program
Define its semantics in terms of fixpoints of that operator
A semantic operator

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

\[
T_\mathcal{P}(I) = \{ p \mid \exists r \in \mathcal{P} : \text{head}(r) = p \land I \models \text{body}(r) \}.
\]

- Intuitively: update \( I \) according to \( \mathcal{P} \).
- Already defined by van Emde 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

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

**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
  \[
  t = (t, t) \quad f = (f, f) \quad u = (f, t) \quad i = (t, f)
  \]
- First: lower bound; second: upper bound
- Alternative view: interval in \( f \leq t \)
- Alternative representation of four-valued interpretation: tuple \((l_1, l_2)\) of two two-valued interpretations
- \( l_1 \) all certainly true atoms; \( l_2 \) all possibly true 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

$$I = \text{lfp } \Psi_P(\cdot, I)$$

- $\Psi_P(\cdot, I)$ is immediate consequence operator of reduct $\mathcal{P}^I$.
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. $\forall v : O(x \land v) \leq v \Rightarrow x \leq v$. 
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):

  \[ \text{Bird}(x) \land M \text{ Fly}(x) \Rightarrow \text{Fly}(x) \]

- Autoepistemic logic (AEL):

  \[ \text{Bird}(x) \land \neg K \neg \text{Fly}(x) \Rightarrow \text{Fly}(x) \]

- 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?
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

1 History & Origin

2 Further theoretical advances

3 Application Domains
   - Weighted Abstract Dialectic Frameworks

4 Future work

5 Conclusion
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:

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

- 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

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 \(i < \beta\);
- \((x_\lambda, y_\lambda) = \text{lub}_{\leq_p} \{(x_i, y_i) \mid i < \lambda\}\) 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 \( O(x \land v) \leq v \), 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: $\land = \cap$ and $\lor = \cup$

Definition (Grounded)

We call $x \in L$ grounded for $O$ if for each $v \in L$ such that $O(x \land v) \leq v$, 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

\[
\begin{align*}
\{ & p \leftarrow p. \\
q & \leftarrow \neg p \lor q. \\
\}
\end{align*}
\]
## Approximation Fixpoint Theory

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

### 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. \( \forall v : O(x \land v) \leq v \Rightarrow x \leq v \).
Relationships between fixpoints

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

1. History & Origin

2. Further theoretical advances

3. Application Domains
   - Weighted Abstract Dialectic Frameworks

4. Future work

5. Conclusion
Extensions of logic programming

- Semantics for syntactic extensions
  - Aggregates in bodies
  - HEX atoms
  - Integration of description logics with logic programming
  - Higher-order logic programming
Given a logic program with aggregates in the body, e.g.,

\[
controls(C_1, C_2) \leftarrow \sum_{\{C_3, N \mid (controls(C_1, C_3) \lor C_1 = C_3) \land OwnsShares(C_3, C_2, N)\}} N \geq 50
\]

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
Active Integrity Constraints (AICs)

- Modern-day databases: *integrity constraints* essential
- What if such constraints are violated? How to fix the database?
- Active integrity constraints: rules of the form
  \[ p \land q \land \neg r \supseteq \neg p \]
- 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

\[ \neg a \land \neg c \]

\[ \neg b \land \neg a \]

\[ a \rightarrow b \]

\[ b \rightarrow c \]

\[ c \rightarrow a \]
Abstract Dialectical Frameworks (ADFs)

$\neg a \lor \neg c$

Diagram:
- Nodes: $a$, $b$, $c$, $f$
- Edges: $a \rightarrow b$, $b \rightarrow c$, $c \rightarrow a$
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

\[ \max\{a, c\} \]

\[ \max\{a, b\} \]

Diagram:
- Node a connected to b and c with an edge labeled 0.25.
- Node b connected to c with an edge labeled \( \max\{a, c\} \).
- Node c connected to a with an edge labeled \( \max\{a, b\} \).
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 \( u \) 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 \( w \)-ADF\( s \)? 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
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

\[
\max\{a, 1 - c\}
\]

\[
\max\{a, 1 - b\}
\]
wADFs: AFT-style Semantics

Over the unit interval

- *(Stable) Models*: assignments \( I \) s.t.
  \[ I(a) = 0.25 \text{ and } I(b) = 1 - I(c) \in [0.25, 0.75]. \]
- *Grounded interpretation* assigns 0.25 to \( a \) and
  \([0.25, 0.75]\) to \( b \) and \( c \),
A weighted abstract dialectical framework over \( \nu \) 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:

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

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

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

$\nu$-wADF = a tuple $(S, C)$ with:

- $S$ a vocabulary
- $C$ a family acceptance functions $C^i_s : \text{int}(\nu, S) \rightarrow \nu$.

BSWW (AAAI’18):

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

AFT-style:

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

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 distinction between values and approximations
- Complexity analysis
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?
1 History & Origin

2 Further theoretical advances

3 Application Domains
   • Weighted Abstract Dialectic Frameworks

4 Future work

5 Conclusion
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. $\forall v : O(x \land v) \leq v \Rightarrow x \leq v$. 