Fixpoint Semantics for Logic Programming

See Melvin Fitting 2002
If you read and understand Fitting’s survey paper you have learned a sufficient amount of knowledge in this class.

Note that some things are given a slightly different name – but mean the same as things we have learned here.
Student evaluation
Simplifications

Given a logic program P with clauses C, Construct P* with clauses C* by
- replace „A ← .“ by „A ← true“, 
ground instantiate all clauses from C, 
if the ground atom A is not the head of any member of P*, add „A ← false“. 

Example:

P(x) :- Q(x), R(x).
R(a).

Becomes P*:

R(a) :- true.
P(a) :- Q(a), R(a).
Q(a) :- false.
Truth ordering

Minimize with respect to order, i.e. default to false:

Definition: The space \{false, true\} is given the truth ordering false \( <_t \) true, with \( x <_t y \) not holding in any other case. We use \( \leq_t \) as usual for \( <_t \) or \( = \).

\[
\begin{array}{ccc}
\text{false} & <_t & \text{true} \\
\end{array}
\]

This ordering is extended to interpretations pointwise:

\( I_1 \leq_t I_2 \) if and only if \( I_1(A) \leq_t I_2(A) \) for all ground atoms \( A \).
Side remark

$TP_{\omega}$ is not necessarily the biggest fixpoint, but

$TP_{\alpha}$ for some $\alpha > \omega$
Fixpoints

We know: Normal programs do not have one smallest fixpoint

Approach:
1. Consider two (or more) fixpoints
2. Consider multi-valued interpretations
Partial interpretations

We know: A classical interpretation assigns every ground atom a truth value from \{true, false\}.

Consider:

\[
P :- P, Q.
\]

Smallest fixpoint: \{Q\}

Largest fixpoint: \{Q, P\}

Idea:

What is true in both fixpoints is true.
What is true in one fixpoint, but false in the other is uncertain \perp.
Partial interpretation

**Definition:** A partial valuation is a mapping $I$ from the set of ground atoms to the set $\{\bot, \text{false}, \text{true}\}$, meeting the conditions

\[
I(\text{false}) = \text{false}
\]

and

\[
I(\text{true}) = \text{true}
\]

We often refer to partial valuations as three valued.
Three valued knowledge ordering

**Definition:** The space \{\bot, \text{false}, \text{true}\} is given a knowledge ordering \(\bot <_k \text{false}, \bot <_k \text{true}\), with \(x <_k y\) not holding in any other case. Then \(\leq_k\) is defined as usual.

\[\begin{align*}
\text{false} & \quad <_k \quad \bot & \quad <_k \quad \text{true} \\
\end{align*}\]

The ordering is again extended to partial interpretations pointwise:

\[I_1 \leq_k I_2 \iff I_1(A) \leq_k I_2(A)\] for all ground atoms \(A\).
Alternative notation

Describe three-valued interpretation $I$ as pair $(T,F)$ of true ground atoms $T$ and false ground atoms $F$.

Then $I_1 \leq_k I_2$ iff $T_1 \subseteq T_2$ and $F_1 \subseteq F_2$ ("$I_2$ knows more than $I_1$"")
Mapping $\Phi_P$

**Definition.** Let $P$ be a normal program. An associated mapping $\Phi_P$, from partial interpretations to partial interpretations, is defined as follows.

$$\Phi_P(I)=J$$

where $J$ is the unique partial interpretation determined by the following: for a ground atom $A$,

1. $I(A)=true$ if there is a general ground clause $A \leftarrow B_1, \ldots, B_n$ in $P^*$ with head $A$, such that $I(B_1)=true$ and $\ldots$ and $I(B_n)=true$.

2. $I(A)=false$ if, for every general ground clause $A \leftarrow B_1, \ldots, B_n$ in $P^*$ with head $A$, $I(B_1)=false$, or $\ldots$, $I(B_n)=false$.

3. $I(A)=\bot$ otherwise.
Kleene’s strong three-valued logic

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A ∨ B</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>⊥</td>
<td>⊥</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>⊥</td>
<td>false</td>
</tr>
<tr>
<td>⊥</td>
<td>true</td>
<td>⊥</td>
</tr>
<tr>
<td>⊥</td>
<td>false</td>
<td>⊥</td>
</tr>
<tr>
<td>⊥</td>
<td>⊥</td>
<td>⊥</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A ∨ B</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>⊥</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>⊥</td>
<td>false</td>
</tr>
<tr>
<td>⊥</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>⊥</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>⊥</td>
<td>⊥</td>
<td>⊥</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>¬ A</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>True</td>
</tr>
<tr>
<td>⊥</td>
<td>⊥</td>
</tr>
</tbody>
</table>
Monotonicity of $\Phi_P$

**Proposition:** For a general program $P$, the operator $\Phi_P$ is monotone with respect to $\leq_k$:

$I_1 \leq_k I_2$ implies $\Phi_P(I_1) \leq_k \Phi_P(I_2)$.

**Note:** The smallest fixed point of $\Phi_P$ supplies the Fitting semantics (also called Kripke-Kleene semantics) with

\[
\begin{align*}
\Phi_P \uparrow 0 &= \bot \\
\Phi_P \uparrow \alpha + 1 &= \Phi_P(\Phi_P \uparrow \alpha) \\
\Phi_P \uparrow \lambda &= \bigcup \{ \Phi_P \uparrow \alpha \mid \alpha < \lambda \}
\end{align*}
\]

with $\lambda$ being a limit ordinal, but $\bigcup$ is with respect to $\leq_k$.
Differences and Commonalities between $T_P$ and $\Phi_P$

Q :- Q.
Fixpoint for $T_P$ is $\{\}$, i.e. $I(Q) = \text{false}$

Q :- not Q.
No fixpoint.

Q :- Q.
Fixpoint for $\Phi_P$ is ($\{\}$,$\{\}$), i.e. $I(Q) = \perp$.

Q :- not Q.
Fixpoint for $\Phi_P$ is ($\{\}$,$\{\}$), i.e. $I(Q) = \perp$.

**Proposition**: Let $P$ be a definite program. Let $I_k$ be the smallest fixed point of $\Phi_P$ (with respect to $\leq_k$), and let $J_t$ and $J_t$ be the smallest and the biggest fixed points of $T_P$ (with respect to $\leq_t$). Then, for a ground atom $A$,

1. If $J_t(A) = J_t(A)$, then $I_k(A)$ has this common value.
2. If $J_t(A) \neq J_t(A)$ then $I_k(A) = \perp$. 
Belnap’s four-valued Logic

Knowledge and truth ordering

Default f: closed world, default ⊥: open world
Truth values for Belnap’s logic

$\bot = \{\}$
false = \{false\}
true={true}
$\top=$\{true,false\}

$\leq_k$ is now simply defined by $\subseteq$ over $I=(T,F)$

$\leq_k$ is a lattice, $\leq_t$ is a lattice; their combination is a bi-lattice.
Operations

Logical connectives formalizable as (infinitely distributive) functions on this ordering:

- $a \lor b = \text{sup}_t(a, b)$
- $a \land b = \text{inf}_t(a, b)$
- $a \oplus b = \text{sup}_k(a, b)$  
  
  „gullibility“
- $a \otimes b = \text{inf}_k(a, b)$  
  
  „consensus“
- $\neg a = \begin{cases} 
  f, & \text{if } a = t \\
  t, & \text{if } a = f \\
  a, & \text{otherwise} 
\end{cases}$

Four binary operations, all distributive laws hold.
**Newly define interpretations**

\[ I(A \land B) = I(A) \land I(B) \]
\[ I(A \otimes B) = I(A) \otimes I(B) \]
etc.

**Definition.** Let \( P \) be a normal program. Let \( P^* \) be its grounding as defined before. Let \( P^{**} \) be the completion of \( P^* \) (with possibly infinitely long ground clauses).

\[ \Phi_P(I) = J, \]

where \( J \) is the unique interpretation determined by the following:

if \( A \leftarrow B \) is in \( P^{**} \), then \( J(A) = I(B) \),

where we use Belnap's logic to evaluate \( I(B) \).
Smallest and biggest fixed points

Proposition 19: Let $i_t$ and $l_t$ be the smallest and biggest fixed points of the four-valued operator $\Phi_P$ with respect to the $\leq_t$ ordering, where $P$ is a definite program. Likewise, let $j_k$ and $J_k$ be the smallest and biggest fixed points of $\Phi_P$ with respect to the $\leq_k$ ordering. We can state that:

\[
j_k = i_t \otimes l_t
\]

\[
J_k = i_t \oplus l_t
\]

\[
i_t = j_k \land J_k
\]

\[
l_t = j_k \lor J_k
\]
On the Semantics of Trust on the Semantic Web

Simon Schenk
ISWC 2008, Karlsruhe, Germany
To judge, whether Quantum of Solace is a good action movie, we need *paraconsistent* reasoning:

\[
\text{olga:GoodActor} \rightarrow T \quad \text{qos:GoodAction} \rightarrow t
\]
ISWeb - Information Systems & Semantic Web
Steffen Staab
staab@uni-koblenz.de  
24
Other Examples

Collaborative Ontology Editing
- Editors trusted differently
- Personal relation
- Even if possible, strict trust order for employees might be illegal

Caching
- Distinguish between certain and possibly outdated information

...
Overview

- Motivation
- Logical Bilattices
- „Trust Bi-Lattices“
- SROIQ on bilattices
- Outlook and Conclusion
Logical Bi-lattices

Knowledge and truth ordering

Logical connectives formalizable as (infinitely distributive) functions on this ordering:

- \( a \vee b = \sup_t(a, b) \)
- \( a \wedge b = \inf_t(a, b) \)
- \( a \oplus b = \sup_k(a, b) \)
- \( a \otimes b = \inf_k(a, b) \)
- \( \neg a = t, \) if \( a = f \)
- \( a, \) otherwise

Default \( f: \) closed world, default \( \bot: \) open world
Other bilattices

e.g. *designed* for default reasoning
**Approach**

**Generate** logical bilattice based on trust order

Lukasiewicz:

Derive (distributive) bilattice from two (distributive) lattices as follows:

Given two distributive lattices $L_1$ and $L_2$, create a bilattice $L$, where the nodes have values from $L_1 \times L_2$, such that

- $(a,b) \leq_k (x,y)$ iff $a \leq_{L_1} x \land b \leq_{L_2} y$
- $(a,b) \leq_t (x,y)$ iff $a \leq_{L_1} x \land y \leq_{L_2} b$

For example, $\text{FOUR} = (0,t) \times (0,f)$:

![Diagram of FOUR bilattice]
Generate L_1 and L_2 from trust order T over information sources S:

\[ L_1 = L_2 = \{(f_i, t_i) \mid i \in S\} \cup \{(t_i, t_j) \mid (i, j) \in T\} \cup \{(f_i, f_j) \mid (j, i) \in T\} \]
Augmented Trust Order

Derive $L_1$ and $L_2$ from *augmented* trust order $T$ over information sources $S$:

$L_1 = L_2 = \{(f_i, t_i) \mid i \in S\} \cup \{(t_i, t_j) \mid (i,j) \in T\} \cup \{(f_i, f_j) \mid (j,i) \in T\}$
Use trust order to derive a logical bilattice.

Example for comparable information sources:
FOUR-T (2)

leads to:

a) comparable sources

b) incomparable sources
Extending SROIQ to Logical Bi-lattices

Interpretation in SROIQ:
- Class: set of individuals
- Property: relation (set of pairs)

Interpretation in SROIQ-T:
- Class: function from individuals to truth values
- Property: function from pairs of individuals to truth values

From SROIQ to SROIQ-T
- Replace intersection and union by conjunction and disjunction
- Replace "for all" and "exists" by conjunction and disjunction over all individuals

\[(C_1 \cap C_2)^I = C_1 \cap C_2\]
\[(\exists R.C)^I = \{ x: (x,y) \in R^I \land y \in C^I \}\]

Satisfiability: There exists a model, which makes all axioms true

\[(C_1 \cap C_2)^I(x) = C_1^I(x) \land C_2^I(x)\]
\[( \exists R.C)^I(x) = \bigvee_{y \in \Delta^I} R^I(x,y) \land C^I(y)\]

Satisfiability wrt. Truth value \(u\): There exists a model, which assigns a truth value \(\geq u\) to all axioms
SROIQ on Bilattices (1)

\[ \top^I(x) = \top_{yy}, \text{where } y \text{ is the information source, defining } \top^I(x) \]

\[ \bot^I(x) = \bot_{yy}, \text{where } y \text{ is the information source, defining } \bot^I(x) \]

\[ (C_1 \cap C_2)^I(x) = C_1^I(x) \land C_2^I(x) \]

\[ (C_1 \cup C_2)^I(x) = C_1^I(x) \lor C_2^I(x) \]

\[ \neg C^I(x) = \neg C^I(x) \]

\[ (S^-)^I(x, y) = S^I(y, x) \]

\[ (\forall R.C)^I(x) = \bigwedge_{y \in \Delta^I} R^I(x, y) \rightarrow C^I(y) \]

\[ (\exists R.C)^I(x) = \bigvee_{y \in \Delta^I} R^I(x, y) \land C^I(y) \]

\[ (\exists R.\text{Self})^I(x) = R^I(x, x) \]

\[ (\geq nS)^I(x) = \bigvee \{y_1, \ldots, y_m \subseteq \Delta^I, m \geq n \} \bigwedge_{i=1}^{n} S^I(x, y_i) \]

\[ (\leq nS)^I(x) = \neg \bigvee \{y_1, \ldots, y_{n+1} \subseteq \Delta^I \} \bigwedge_{i=1}^{n+1} S^I(x, y_i) \]

\[ \{a_1, \ldots, a_n\}^I(x) = \bigvee_{i=1}^{n} a_i^I = x \]
SROIQ on Bilattices (1)

\[
(R \sqsubseteq S)^T = \bigwedge_{x, y \in \Delta^T} R^T(x, y) \rightarrow S^T(x, y)
\]

\[
(R = S)^T = \bigwedge_{x, y \in \Delta^T} R^T(x, y) \equiv S^T(x, y)
\]

\[
(R_1 \circ \ldots \circ R_n \sqsubseteq S)^T = \bigwedge_{(x_1, x_{n+1}) \in \text{dom}(S^T)} \bigvee_{\{x_2, \ldots, x_n\}} \bigwedge_{i=1}^n R_i^T(x_i, x_{i+1})
\]

\[
(\text{Asy}(R))^T = \bigwedge_{x, y \in \Delta^T} \lnot (R^T(x, y) \land R^T(y, x))
\]

\[
(\text{Ref}(R))^T = \bigwedge_{x \in \Delta^T} R^T(x, x)
\]

\[
(\text{IrR}(R))^T = \bigwedge_{x \in \Delta^T} \lnot R^T(x, x)
\]

\[
(\text{Dis}(R, S))^T = \bigwedge_{x, y \in \Delta^T} R^T(x, y) \rightarrow \lnot S^T(x, y)
\]

\[
(C \sqsubseteq D)^T = \bigwedge_{x \in \Delta^T} C^T(x) \rightarrow D^T(x)
\]

\[
(a : C)^T = C^T(a^T)
\]

\[
((a, b) : R)^T = R^T(a^T, b^T)
\]

\[
a \approx b = a^T = b^T
\]

\[
a \not\approx b = a^T \neq b^T
\]
Application: Inconsistency Resolution

Reasons for Inconsistencies:
\[ tv(a) = t_x : a \leftarrow A \]
\[ tv(a) = f_y : a \leftarrow B \]

\[ f_x \land t_y = \top_{xy} \] (inconsistent)

Subscript of \( \top \) reflects the maximally and minimally trusted information sources, which cause the inconsistency.

Possible resolution: Find minimal inconsistent subontology
Drop minimally trusted axioms.
Application: Inconsistency Resolution

\[ \text{olga:GoodActor} \rightarrow T_{W,SO} \]
\[ \text{qos:GoodAction} \rightarrow T_{M,SO,W} = f_M \oplus t_{SO,W} \]

Minimally and maximally trusted source contributing to the inconsistency

Drop minimally trusted axioms

Not possible for \text{olga:GoodActor}!
Conclusion and Future Work

- Go watch „Quantum of Solace“ (Simon’s recommendation)

- Trust based reasoning on logical bilattices
  - Derived from any partial trust order
  - Applicable to a broad variety of languages

- Current Work:
  - Operationalization
    - Ask Renata!