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.
**Simplifications**

Given a logic program $P$ with clauses $C$,

Construct $P^*$ with clauses $C^*$ by

- replace "$A ← \ldots$" 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 $P^*$

$$R(a) :- true.$$
$$P(a) :- Q(a), R(a).$$
$$Q(a) :- false.$$
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 \(=\).

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

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

$\top_P \downarrow_\omega$ is not necessarily the biggest fixpoint, but $\top_P \downarrow_\alpha$ 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.
**Definition**: A partial valuation is a mapping $I$ from the set of ground atoms to the set \{$\bot$, false, true$\}$, meeting the conditions

$I(false) = false$

and

$I(true) = true$

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

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

\[
\begin{array}{ccc}
\text{false} & \xrightarrow{<_k} & \text{true} \\
\downarrow & & \downarrow \\
⊥ & & ⊥ \\
\end{array}
\]

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. \( J(A) = \text{true} \) if there is a general ground clause \( A \leftarrow B_1, \ldots, B_n \) in \( P^* \) with head \( A \), such that \( I(B_1) = \text{true} \) and \( \ldots \) and \( I(B_n) = \text{true} \).
2. \( J(A) = \text{false} \) if for every general ground clause \( A \leftarrow B_1, \ldots, B_n \) in \( P^* \) with head \( A \), \( I(B_1) = \text{false} \), or \( \ldots \), \( I(B_n) = \text{false} \).
3. \( J(A) = \bot \) 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>⊥</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>¬ 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>
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

$$\Phi_P \uparrow 0 = \perp$$
$$\Phi_P \uparrow \alpha + 1 = \Phi_P(\Phi_P \uparrow \alpha)$$
$$\Phi_P \uparrow \lambda = \bigcup \{\Phi_P \uparrow \alpha \mid \alpha < \lambda\}$$

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 $\emptyset$, i.e. $I(Q)=\text{false}$

Q :- Q.
Fixpoint for $\Phi_P$ is $(\emptyset,\emptyset)$, i.e. $I(Q)=\bot$.

Q :- not Q.
No fixpoint.

Q :- not Q.
Fixpoint for $\Phi_P$ is $(\emptyset,\emptyset)$, i.e. $I(Q)=\bot$.

**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 $A$,

- If $j_t(A) = J_t(A)$, then $I_k(A)$ has this common value.
- If $j_t(A) \neq J_t(A)$ then $I_k(A)=\bot$. 


Belnap’s four-valued Logic

Knowledge and truth ordering

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

⊥ = {}
false = {false}
true={true}
⊤={true,false}

≤_k is now simply defined by ⊆ over I=(T,F)

≤_k is a lattice, ≤_t is a lattice; their combination is a bi-lattice.
Logical connectives formalizable as (infinitely distributive) functions on this ordering:

\[ a \lor b = \sup_t(a,b) \]
\[ a \land b = \inf_t(a,b) \]
\[ a \oplus b = \sup_k(a,b) \]
\[ a \otimes b = \inf_k(a,b) \]

\[ \neg a = \begin{cases} 
  f, & \text{if } a = t \\
  t, & \text{if } a = f \\
  a, & \text{otherwise}
\end{cases} \]

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).

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

\begin{align*}
    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
\end{align*}
On the Semantics of Trust on the Semantic Web

Simon Schenk
ISWC 2008, Karlsruhe, Germany
"Quantum of Solace"

"Olga Kurylenko toughest Bond-Girl ever."
olga: GoodActor
qos: GoodAction

WELT ONLINE
"Olga Kurylenko flat like a stale Martini."
olga: ¬GoodActor
qos: GoodAction

Spiegel ∪ Welt globally inconsistent.

To judge, whether Quantum of Solace is a good action movie, we need paraconsistent reasoning:

olga: GoodActor → ⊤ qos: GoodAction → ⊤
"Quantum of Solace"

**SPIEGEL ONLINE**
- olga: GoodActor
- qos: GoodAction

**WELT ONLINE**
- olga: ¬GoodActor
- qos: GoodAction

**Mail Online**
- olga: ¬GoodActor
- qos: ¬GoodAction

**DIE ZEIT**

**FAZ.NET**

**SPIEGEL ONLINE**

**Mail Online**

Trust in News Sources

qos: GoodAction $\rightarrow$ t_{so,w}

olga: GoodActor $\rightarrow$ t_{so,w,m}

General Trust Order

daniel: GoodActor

?
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:

\[ a \vee b = \sup_{k}(a,b) \]
\[ a \wedge b = \inf_{t}(a,b) \]
\[ a \oplus b = \sup_{k}(a,b) \]
\[ a \otimes b = \inf_{t}(a,b) \]

\[ \neg a = \begin{cases} f, & \text{if } a = t \\ t, & \text{if } a = f \\ a, & \text{otherwise} \end{cases} \]

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

- e.g. *designed* for default reasoning

\[
\text{SEVEN}
\]

\[
\text{NINE}
\]
**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, FOUR = $\{0,t\} \times \{0,f\}$:

```
(0,0) -> (0,f) -> (t,0) -> (t,f) -> (0,0)
```

```
FOUR
```

Generate Lattice from Trust Order

Derive $L_1$ and $L_2$ from trust order $T$ over information sources $S_i$:

$L_1 = L_2 = \{ (f_i, t_i) \mid (f_i, t_i) \in S \} \cup \{(t_i, t_j) \mid (i, j) \in T \} \cup \{(f_i, f_j) \mid (j, i) \in T \}$

Problem: $t_b \oplus t_c = ? t_\infty$
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
Application: Inconsistency Resolution

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

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

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

Minimally and maximally trusted source contributing to the inconsistency

MUPS for qos:GoodAction
qos:GoodAction $\rightarrow$ $t_{SO,W}$
qos:GoodAction $\rightarrow$ $f_M$
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:
  - Publication almost accepted:
    - Ontology Debugging Using Provenance