Bilattices
A Survey

Melvin Fitting

City University of New York

comet.lehman.cuny.edu/fitting

Basics

A *bilattice* is a set with two lattice orderings that are interconnected somehow.

(I'll assume lattices are complete, for this talk.)

Question: interconnected how?

Pre-Bilattices

A *pre-bilattice* is $\mathcal{B} = \langle \mathcal{B}, \leq_t, \leq_k \rangle$ where $\mathcal{B}$ is a non-empty set and $\leq_t$ and $\leq_k$ are lattice orderings.

$\leq_t$ is a "truth" ordering, $\leq_k$ is an "information" ordering.

Matt Ginsberg

Introduced bilattices in 1988.

Connection between lattices was through negation.

I need something stronger.

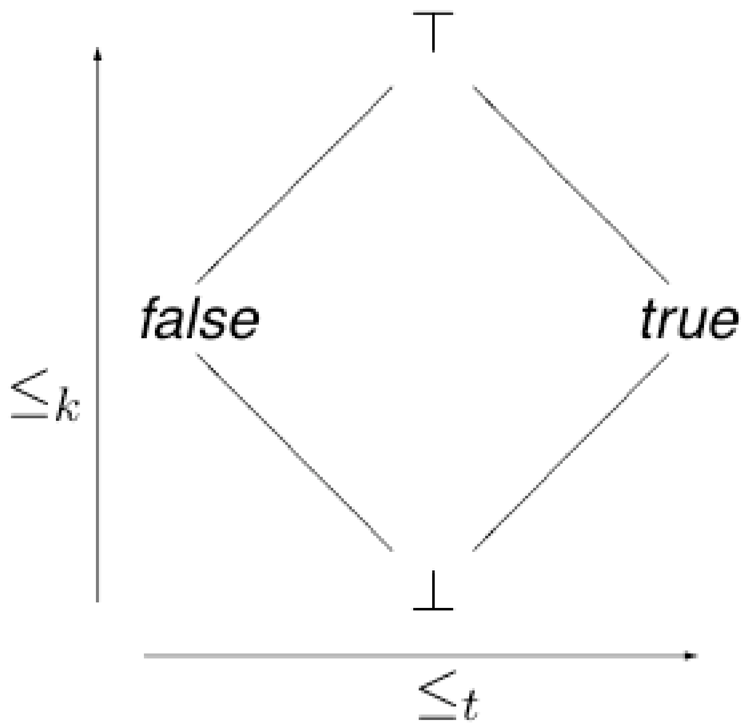
Lattice Operations

Under $\leq_t$: meet $\wedge$, join $\vee$, infinite meet $\bigwedge$, infinite join $\bigvee$, smallest *false*, largest *true*

Under $\leq_k$: meet $\otimes$, join $\oplus$, infinite meet $\prod$, infinite join $\sum$, smallest $\bot$, largest $\top$

Assume *false*, *true*, $\bot$, $\top$ all different.

Simplest Example



Belnap's four-valued logic.

Bilattice Conditions

Interlaced: All (finitary and infinitary) operations monotone with respect to both orderings. Example: $a \leq_t b$ implies $a \oplus c \leq_t b \oplus c$.

Distributive: All distributive laws hold. Example: $a \wedge (b \oplus c) = (a \wedge b) \oplus (a \wedge c)$

Distributive implies Interlaced.

Belnap example is distributive.

Negation and Conflation

Negation operation, $\neg$: reverses the $\leq_t$ ordering, leaves unchanged the $\leq_k$ ordering, and $\neg\neg x = x$.

Conflation operation, $-$, reverses the $\leq_k$ ordering, leaves unchanged the $\leq_t$ ordering, and $- - x = x$.

If a bilattice has both, they must *commute*, $-\neg x = \neg - x$ for all x.

Example: Belnap's again.

Function Spaces

Given: $\mathcal{B}$ a pre-bilattice,

S a non-empty set.

$\mathcal{B}^S$ is the function space. Define $\leq_t$ and $\leq_k$ pointwise. Then $\mathcal{B}^S$ also a pre-bilattice.

$\mathcal{B}^S$ is interlaced if $\mathcal{B}$ is. Similarly for distributive, having a negation, having a conflation.

Example — People

P is a set of people with opinions

Bilattice member (generalized truth value) is $\langle Y, N \rangle$, where $Y \subseteq P$, $N \subseteq P$

$\langle Y_1, N_1 \rangle \leq_k \langle Y_2, N_2 \rangle$ if $Y_1 \subseteq Y_2$ and $N_1 \subseteq N_2$

$\langle Y_1, N_1 \rangle \leq_t \langle Y_2, N_2 \rangle$ if $Y_1 \subseteq Y_2$ and $N_2 \subseteq N_1$

$$\langle Y_1, N_1 \rangle \wedge \langle Y_2, N_2 \rangle =$$
$$\langle Y_1 \cap Y_2, N_1 \cup N_2 \rangle$$

$$\langle Y_1, N_1 \rangle \otimes \langle Y_2, N_2 \rangle =$$
$$\langle Y_1 \cap Y_2, N_1 \cap N_2 \rangle$$

$$\neg \langle Y, N \rangle = \langle N, Y \rangle$$

$$-\langle Y, N \rangle = \langle P - N, P - Y \rangle$$

(One-person case gives Belnap's logic)

Example—Fuzzy

Bilattice member (generalized truth value) is $\langle y, n \rangle$, where $y, n \in [0, 1]$

$\langle y_1, n_1 \rangle \leq_k \langle y_2, n_2 \rangle$ if $y_1 \leq y_2$ and $n_1 \leq n_2$

$\langle y_1, n_1 \rangle \leq_t \langle y_2, n_2 \rangle$ if $y_1 \leq y_2$ and $n_2 \leq n_1$

$$\langle y_1, n_1 \rangle \wedge \langle y_2, n_2 \rangle = \langle \min(y_1, y_2), \max(n_1, n_2) \rangle$$

$$\langle y_1, n_1\rangle \otimes \langle y_2, n_2\rangle =$$
$$\langle \min(y_1, y_2), \min(n_1, n_2)\rangle$$

$$\neg\langle y, n\rangle = \langle n, y\rangle$$

$$-\langle y, n\rangle = \langle 1 - n, 1 - y\rangle$$

(Can be combined with people example to produce “fuzzy people”)

General Construction

Let L_1 and L_2 be complete lattices.

Define a pre-bilattice $L_1 \odot L_2$ by:

Carrier set is $L_1 \times L_2$

$\langle a_1, a_2 \rangle \leq_k \langle b_1, b_2 \rangle$ if

$a_1 \leq_{L_1} b_1$ and $a_2 \leq_{L_2} b_2$

$\langle a_1, a_2 \rangle \leq_t \langle b_1, b_2 \rangle$ if

$a_1 \leq_{L_1} b_1$ and $b_2 \leq_{L_2} a_2$

$L_1 \odot L_2$ always interlaced bilattice

$L_1 \odot L_2$ distributive

if L_1 and L_2 distributive

General Construction Continued

If $L_1 = L_2 = L$, $L \odot L$ has a negation:

$\neg\langle a, b\rangle = \langle b, a\rangle$

If also L has an involution

(order reversing isomorphism)

$L \odot L$ has a conflation:

$-\langle a, b\rangle = \langle \overline{b}, \overline{a}\rangle$

And negation and conflation

will commute.

General Construction Is General

Every interlaced bilattice is isomorphic to $L_1 \odot L_2$ for some lattices L_1 and L_2.

Every distributive bilattice is isomorphic to $L_1 \odot L_2$ for some distributive lattices L_1 and L_2.

Every bilattice with negation is isomorphic to $L \odot L$ for some lattice L.

Every bilattice with commuting negation and conflation is isomorphic to $L \odot L$ for some lattice L with an involution.

Exact and Consistent

In an interlaced (or distributive) bilattice with conflation and negation:

A is *exact* if $A = -A$.
A is *consistent* if $A \leq_k -A$.

In Belnap's bilattice, *exact* $= \{$*true*, *false*$\}$; *consistent* $= \{$*true*, $\perp$, *false*$\}$.

Always, *exact* naturally generalizes classical setting, *consistent* naturally generalizes strong Kleene 3-valued logic.

Another Construction

Take a lattice L.

An *interval* is $[a, b] = \{x \mid a \leq x \leq b\}$
(assume $a \leq b$)

$[a_1, b_1] \leq_k [a_2, b_2]$ if
$[a_2, b_2] \subseteq [a_1, b_1]$ (narrowing)

$[a_1, b_1] \leq_t [a_2, b_2]$ if
$a_1 \leq a_2$ and $b_1 \leq b_2$ (shift upwards)

Intervals are the consistent part of a bilattice.

Using $\{\mathit{false}, \mathit{true}\}$, intervals yield Kleene's strong 3-valued logic.

Using $[0, 1]$, intervals yield a "fuzzy" structure.

The Standard Knaster/Tarski

A map Φ is *monotone* on a lattice if

$a \leq b$ implies $\Phi(a) \leq \Phi(b)$

Theorem: A monotone map on a complete lattice has a smallest (and biggest) fixed point.

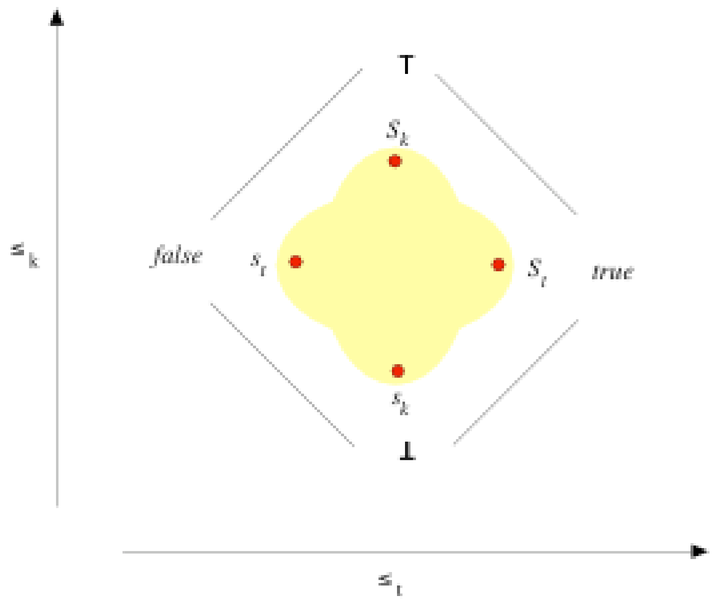
And Still More

Suppose Φ is a bilattice map, monotone in both orders.

Let s_k and S_k be smallest and biggest fixpoints of Φ with respect to $\leq_k$

Let s_t and S_t be smallest and biggest fixpoints of Φ with respect to $\leq_t$

Then Obviously



But More

If bilattice is interlaced:

$$
\begin{aligned}
s_k &= s_t \otimes S_t \\
S_k &= s_t \oplus S_t \\
s_t &= s_k \wedge S_k \\
S_t &= s_k \vee S_k
\end{aligned}
$$

Always Prove Something Simple

Because s_k is least $\leq_k$ fixed point:

$$s_k \leq_k s_t \quad \text{and} \quad s_k \leq_k S_t$$

So of course

$$s_k \leq_k s_t \otimes S_t$$

Continued

Lemma: If $a \leq_t b \leq_t c$ then $a \otimes c \leq_k b$

Proof: Using interlacing,

$$a \otimes b \otimes c \leq_t a \otimes c \otimes c = a \otimes c$$

$$a \otimes b \otimes c \geq_t a \otimes a \otimes c = a \otimes c$$

So

$$a \otimes c = (a \otimes c) \otimes b$$

So

$$a \otimes c \leq_k b$$

Concluded

Because s_t and S_t are extreme $\leq_t$ fixed points

$$s_t \leq_t s_k \leq_t S_t$$

So

$$s_t \otimes S_t \leq_k s_k$$

A Knaster/Tarski Variant

A map Φ is *anti-monotone* on a lattice if $a \leq b$ implies $\Phi(b) \leq \Phi(a)$

Theorem: A monotone map on a complete lattice has extremal alternating fixed points.

a and b (with $a \leq b$) are alternating fixed points if $\Phi(a) = b$ and $\Phi(b) = a$. They are *extremal* if any other pair of alternating fixed points lies between them.

Note: If Φ is anti-monotone, Φ^2 is monotone. Its greatest and least fixed points are extremal alternating fixed points for Φ.

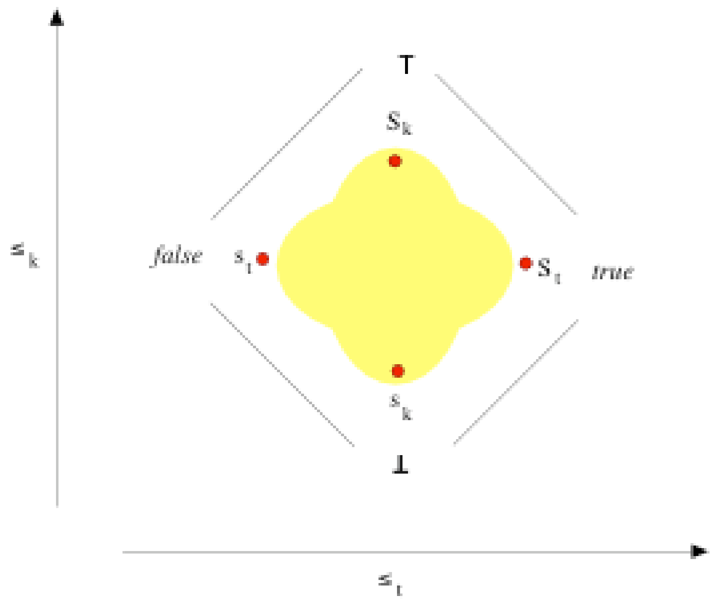
Again Still More

Suppose Φ is a bilattice map, monotone in $\leq_k$ and anti-monotone in $\leq_t$.

Let s_k and S_k be smallest and biggest fixpoints of Φ with respect to $\leq_k$

Let s_t and S_t be smallest and biggest alternating fixpoints of Φ with respect to $\leq_t$

Then Again Obviously



And Again More

If bilattice is interlaced:

$$\begin{aligned} s_k &= s_t \otimes S_t \\ S_k &= s_t \oplus S_t \\ s_t &= s_k \wedge S_k \\ S_t &= s_k \vee S_k \end{aligned}$$

Derived Operators

Suppose $\Psi(x, y)$ maps $\mathcal{B} \times \mathcal{B}$ to $\mathcal{B}$

$\Psi(x, y)$ is monotonic in $\leq_k$

in both arguments

and in $\leq_t$ monotonic in x and

anti-monotonic in y

(Examples later)

For each fixed v, $(\lambda x)\Psi(x, v)$ is monotonic in $\leq_t$, so has a least fixed point.

Ψ' is the map from $\mathcal{B}$ to $\mathcal{B}$ given by

$\Psi'(v)$ is the least (under $\leq_t$) fixed point of $(\lambda x)\Psi(x, v)$

Results

Ψ' is monotonic under $\leq_k$

Ψ' is anti-monotonic under $\leq_t$

Let $\Phi(x) = \Psi(x, x)$. Every fixed point of Ψ' is a fixed point of Φ

Reason: Let s be a fixed point of Ψ'. Then $\Phi(s) = \Psi(s, s) = \Psi(\Psi'(s), s)$ $= \Psi'(s) = s$

Logic Program Clause

$$\text{Head} \leftarrow \text{Body}$$

$$H(\vec{h}) \leftarrow P_1(\vec{p}_1), \ldots P_n(\vec{p}_n),$$
$$\neg Q_1(\vec{q}_1), \ldots, \neg Q_m(\vec{q}_m)$$

In $A(\vec{a})$, A is n-ary relation symbol,
$\vec{a}$ is sequence of terms.

Positive = no negations.

Logic Program

Finite set of clauses (variables allowed in terms).

Positive = no negations.

Note: also extended, disjunctive. Not for now.

Examples

$$E(0) \leftarrow$$
$$E(s(s(x))) \leftarrow E(x)$$

$$E(0) \leftarrow$$
$$E(s(x)) \leftarrow \neg E(x)$$

$$A \leftarrow A$$

$$A \leftarrow \neg A$$

$$A \leftarrow \neg B$$
$$B \leftarrow \neg A$$

Ground Logic Program

All ground instances of a logic program.

Infinite!

Cheating!

Convenient theoretically—assumed from now on.

Example

For:

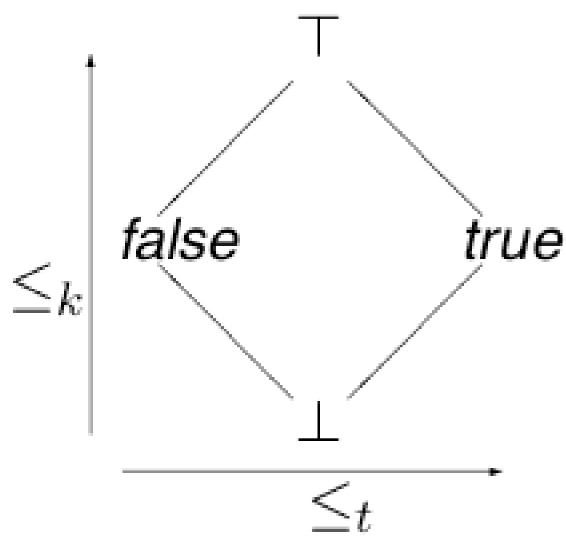
$$
\begin{aligned}
E(0) &\leftarrow \\
E(s(x)) &\leftarrow \neg E(x)
\end{aligned}
$$

Grounded version:

$$
\begin{aligned}
E(0) &\leftarrow \\
E(s(0)) &\leftarrow \neg E(0) \\
E(s(s(0))) &\leftarrow \neg E(s(0)) \\
E(s(s(s(0)))) &\leftarrow \neg E(s(s(0))) \\
&\vdots
\end{aligned}
$$

Valuation

For grounded program $\mathcal{P}$, map v from atoms to Belnap bilattice.



(Other bilattices could be used.)

Valuation on Clause Bodies

$$v(\neg A) = \neg v(A)$$

$$v(\{L_1, \ldots, L_1\}) = v(L_1) \wedge \ldots \wedge v(L_n)$$

$$v(\{\}) = \textit{true}$$

Mapping on Valuations

For grounded program $\mathcal{P}$,

$\Phi_{\mathcal{P}}(v) = w$ where:

$$w(A) = \underline{\bigvee}\{v(B) \mid A \leftarrow B \text{ in } \mathcal{P}\}$$

(Take $\underline{\vee}(\{\}) = \textit{false}$)

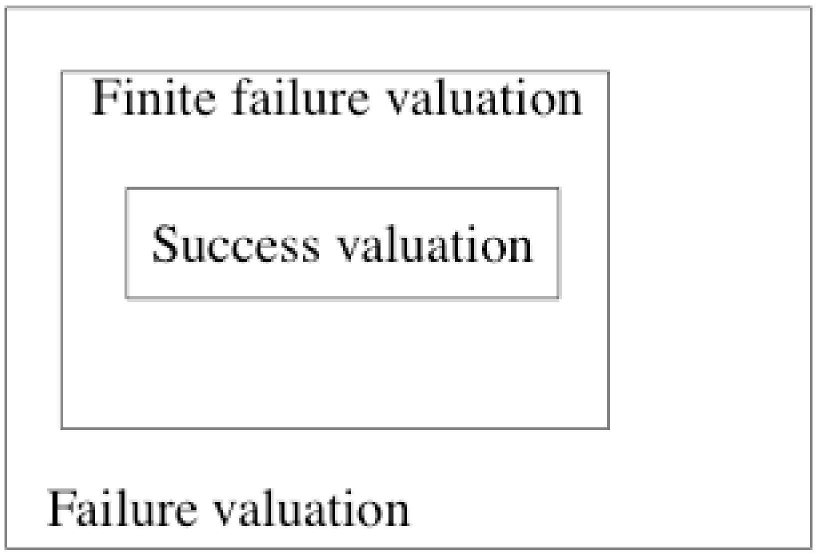
Note: $\Phi_{\mathcal{P}}$ maps *exact* valuations to exact valuations, *consistent* valuations to consistent valuations.

No Negations

If program $\mathcal{P}$ is negation-free $\Phi_{\mathcal{P}}$ is monotone under $\leq_t$.

Least fixpoint is Apt, van Emden, Kowalski semantics (least Herbrand model).

Greatest fixpoint gives *failure* set (not finite failure).



Consider $A \leftarrow A$

But With Negations?

If program $\mathcal{P}$ has negations $\Phi_{\mathcal{P}}$ is not monotone under $\leq_t$.

But is monotone under $\leq_k$.

Least fixpoint gives Kripke-Kleene semantics (more-or-less).

Consider $A \leftarrow \neg A$.

MarkTwain's Suggestion

"Truth is the most precious thing we have, so let us economize it."

Consider

$$Q \leftarrow P$$
$$P \leftarrow P$$

Kripke-Kleene makes Q be $\perp$, but *false* is natural.

Using Two Valuations

Given valuations u and v, define a *pseudo-valuation*

$$(u \triangle v)(A) = u(A)$$
$$(u \triangle v)(\neg A) = \neg v(A)$$

On binary connectives, pseudo-valuations are like valuations.

A Two Input Operator

For grounded program $\mathcal{P}$,
$\Psi_{\mathcal{P}}(u, v) = w$ where:

$w(A) = \underline{\bigvee}\{(u \triangle v)(B) \mid A \leftarrow B \text{ in } \mathcal{P}\}$

$\Psi_{\mathcal{P}}(u, v)$ monotone in u, v under $\leq_k$

$\Psi_{\mathcal{P}}(u, v)$ monotone in u, anti-monotone in v under $\leq_t$

$\Psi_{\mathcal{P}}(u, u) = \Phi_{\mathcal{P}}(u)$

Answer Set Semantics

Look at fixpoints of derived operator $\Psi'_{\mathcal{P}}$

Exact fixed points are *stable* models.

Consistent fixed points are *extended (3-valued)* stable models.

Smallest, under $\leq_k$, is *well-founded* model.

Answer Set Example

There are two exact fixpoints of $\Psi'_{\mathcal{P}}$ for $\mathcal{P}$ being

$$A \leftarrow \neg B$$
$$B \leftarrow \neg A$$

They are v and w where

$$v(A) = \textit{true} \quad w(A) = \textit{false}$$
$$v(B) = \textit{false} \quad w(A) = \textit{true}$$

And By the Way

All this machinery applies to natural language semantics too—Kripke fixpoint semantics.

Approximate Conclusion

Logic programming is alive, and well, and living in Europe.

Answer set semantics is alive, and well, and living in Kentucky and Texas.