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 =.

$$\text{false} \xrightarrow{<_t} \text{true}$$

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

$T_{P\downarrow\omega}$ is not necessarily the biggest fixpoint, but

$T_{P\downarrow\alpha}$ 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 $\perp$.

Partial interpretation

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

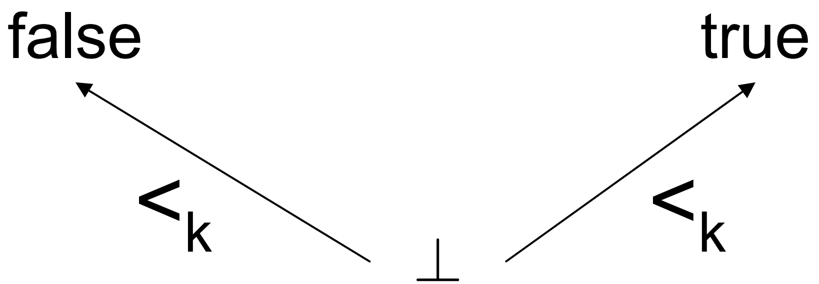
and

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 $\perp <_k$ false, $\perp <_k$ true, with $x <_k y$ not holding in any other case. Then $\leq_k$ is defined as usual.



Complete semi-lattice

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“)

Definition. Let P be a normal program. An associated mapping Φ_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 …. 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 …, $I(B_n)$=false.
3. $I(A)=\perp$ otherwise.

Kleene's strong three-valued logic

A	B	A ∧ B
true	true	true
true	false	false
true	⊥	⊥
false	true	false
false	false	false
false	⊥	false
⊥	true	⊥
⊥	false	false
⊥	⊥	⊥

A	B	A∨ B
true	true	true
true	false	true
true	⊥	true
false	true	true
false	false	false
false	⊥	⊥
⊥	true	true
⊥	false	⊥
⊥	⊥	⊥

A	¬ A
true	false
false	True
⊥	⊥

Monotonicity of Φ_P

<is web>

Proposition: For a general program P, the operator Φ_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 Φ_P supplies the Fitting semantics (also called Kripke-Kleene semantics) with

$$\Phi_P \uparrow 0 = \bot$$

$$\Phi_P \uparrow \alpha+1 = \Phi_P(\Phi_P \uparrow \alpha)$$

$$\Phi_P \uparrow \lambda = \bigcup \{\Phi_P \uparrow \alpha \mid \alpha < \lambda\}$$

with λ being a limit ordinal, but $\bigcup$ is with respect to $\leq_k$

Differences and Commonalities between T_P and Φ_P

Q :- Q.

Fixpoint for T_P is {}, i.e. I(Q)=false

Q :- Q.

Fixpoint for Φ_P is ({},{}), i.e. I(Q)=$\bot$.

Q :- not Q.

No fixpoint.

Q :- not Q.

Fixpoint for Φ_P is ({},{}), i.e. I(Q)=$\bot$.

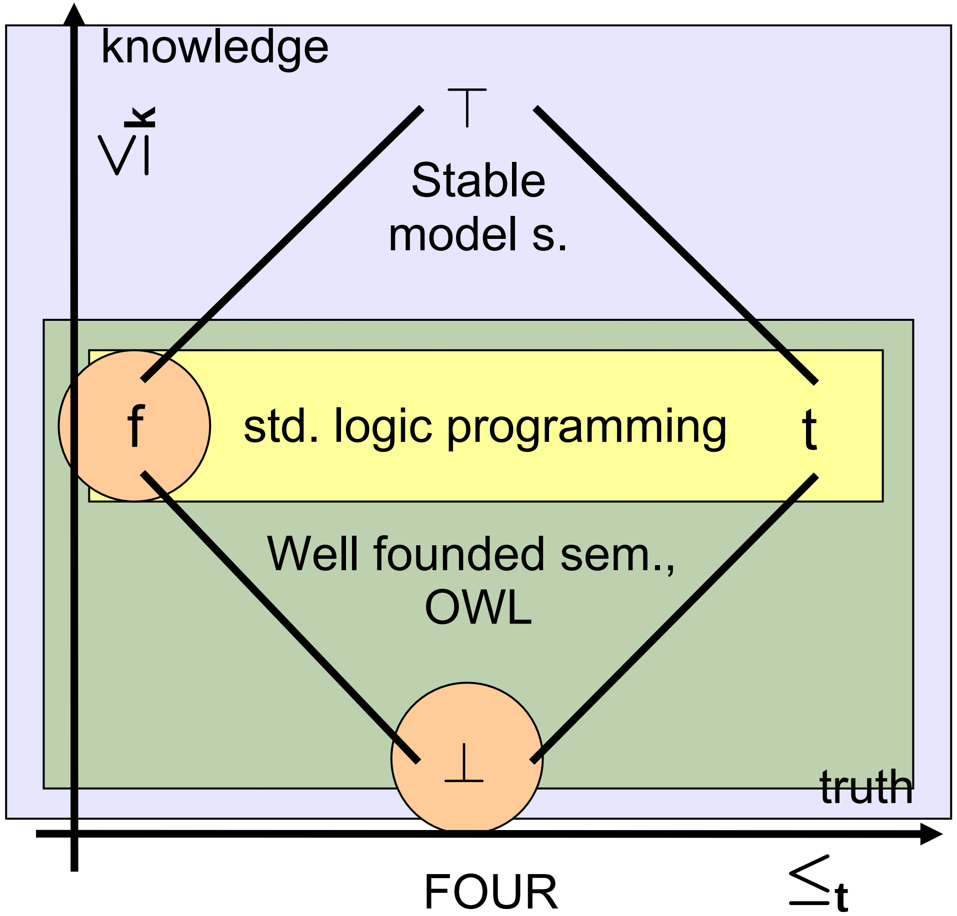
Proposition: Let P be a definite program. Let I_k be the smallest fixed point of Φ_P (with respect to $\leq_k$), and let j_t and J_t be the smallest and the biggest fixed points of T_P (with respect to $\leq_t$). Then, for a ground atom A,

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 $I_k(A)=\bot$.

Belnap‘s four-valued Logic

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}

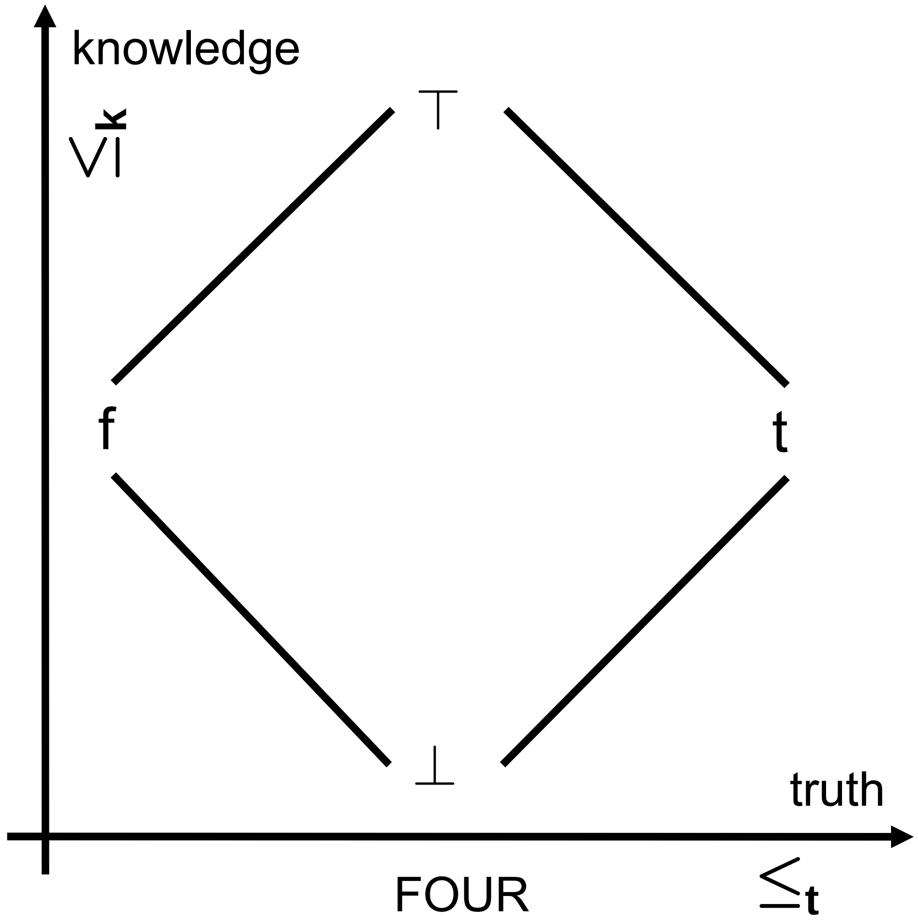
$\leq_k$ is now simply defined by $\subseteq$ over I=(T,F)

$\leq_k$ 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:

- $a \vee b = sup_t(a,b)$
- $a \wedge b = inf_t(a,b)$
- $a \oplus b = sup_k(a,b)$ „gullibility“
- $a \otimes b = inf_k(a,b)$ „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.

$I(A \wedge B) = I(A) \wedge I(B)$

$I(A \otimes B) = I(A) \otimes I(B)$

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).

Proposition 19: Let i_t and I_t be the smallest and biggest vised points of the four-valued operator Φ_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 Φ_P with respect to the $\leq_k$ ordering.

We can state that:

$j_k = i_t \otimes I_t$

$J_k = i_t \oplus I_t$

$i_t = j_k \wedge J_k$

$I_t = j_k \vee J_k$

<is web>

Information Systems & Semantic Web

University of Koblenz ▪ Landau, Germany

On the Semantics of Trust on the Semantic Web

Simon Schenk

ISWC 2008, Karlsruhe, Germany

UNIVERSITÄT KOBLENZ · LANDAU

„Quantum of Solace“

„SPIEGEL ONLINE ghest Bond-Girl ever.“

`olga:GoodActor`

`qos:GoodAction`

WELT ONLINE stale Martini.“

`olga:` $\neg$`GoodActor`

`qos:GoodAction`

`Spiegel` $\cup$ `Welt` globally inconsistent.

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

`olga:GoodActor` $\rightarrow \top$ `qos:GoodAction` $\rightarrow$ `t`

„Quantum of Solace“

SPIEGEL ONLINE

olga:GoodActor

qos:GoodAction

WELT ONLINE

olga:¬GoodActor

Mail Online

¬GoodActor

¬GoodAction

Trust in News Sources

SPIEGEL ONLINE

WELT ONLINE

qos:GoodAction $\rightarrow t_{SO,W}$

olga:GoodActor $\rightarrow \top_{SO,W,M}$

Mail Online

General Trust Order

DIE ZEIT

Frankfurter Allgemeine FAZ.NET

SPIEGEL ONLINE

Mail Online

FOCUS ONLINE

?

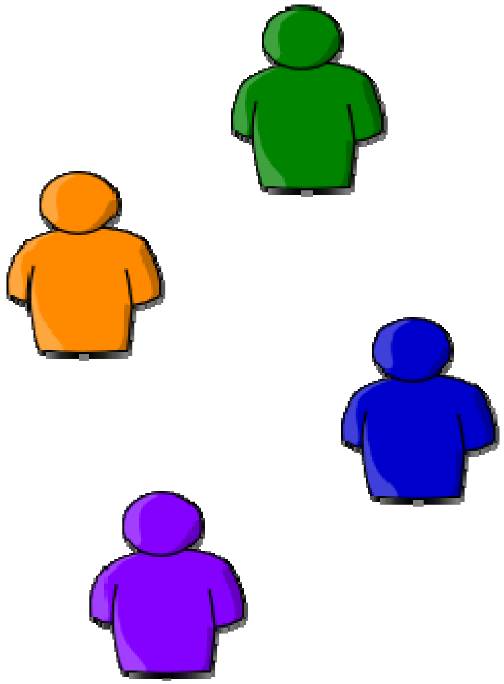
Süddeutsche Zeitung

WELT ONLINE

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

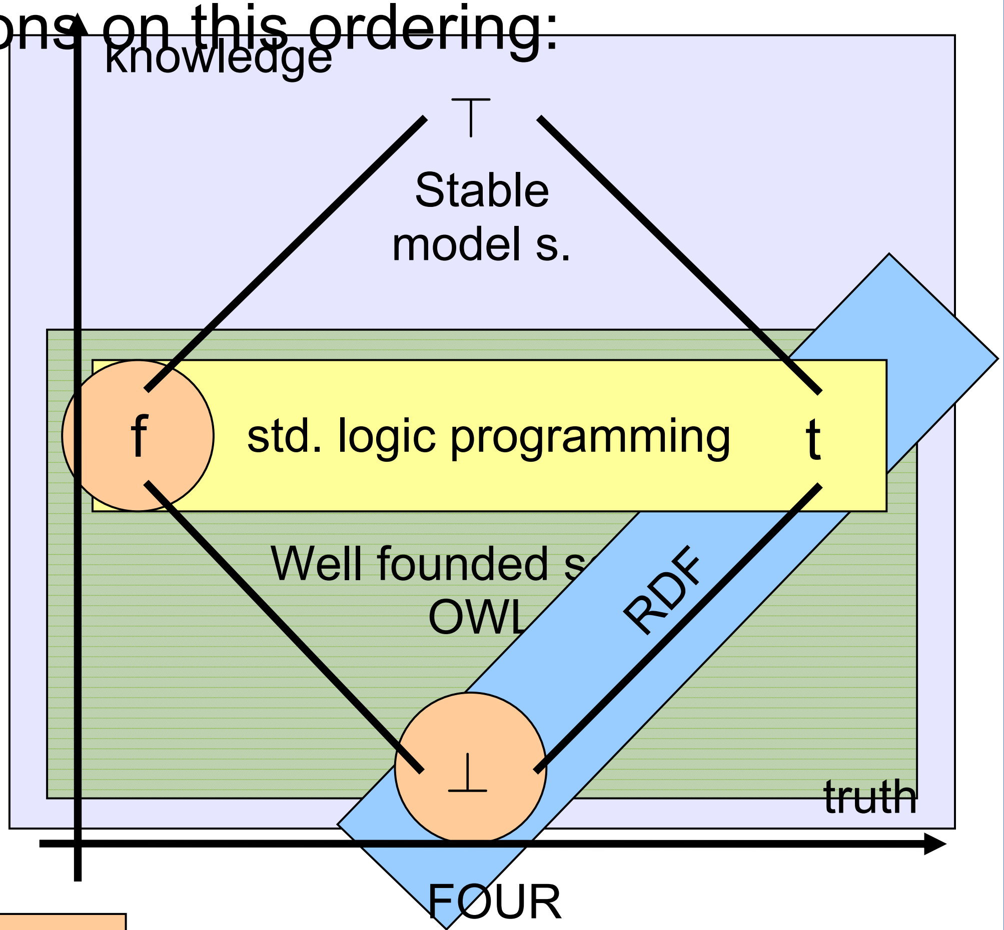
- Motivation
- Logical Bilattices
- „Trust Bi-Lattices“
- SROIQ on bilattices
- Outlook and Conclusion

Logical Bi-lattices

Knowledge and truth ordering

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

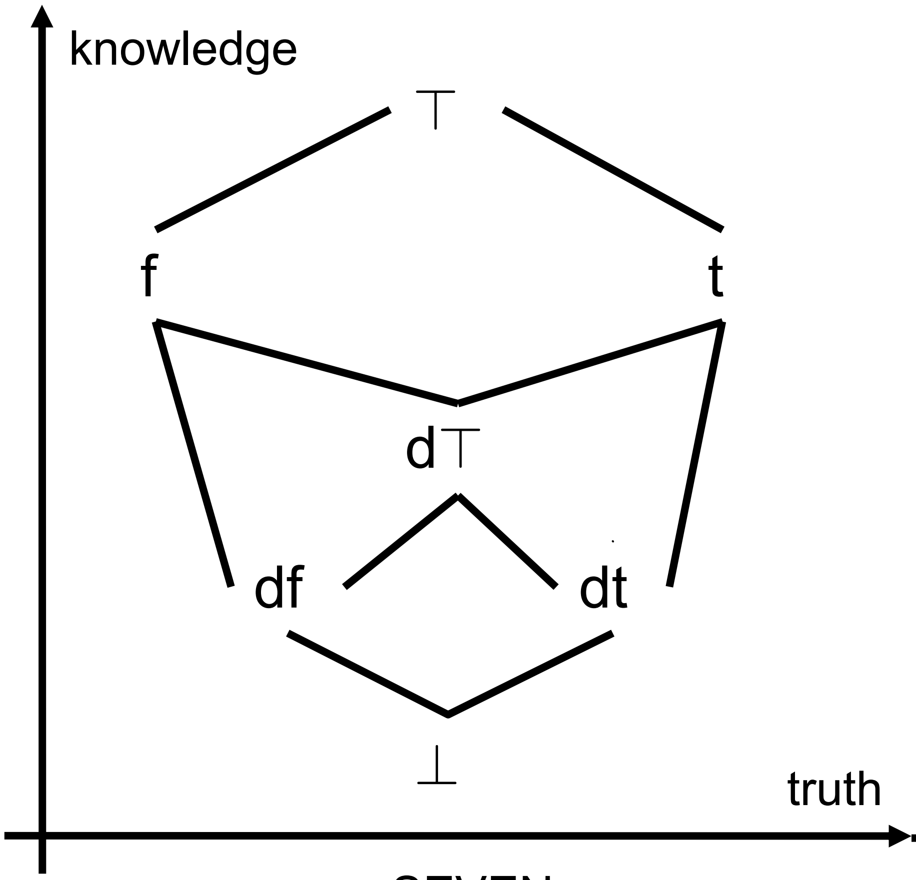
- $a \vee b = sup_t(a,b)$
- $a \wedge b = inf_t(a,b)$
- $a \oplus b = sup_k(a,b)$
- $a \otimes b = inf_k(a,b)$
- $\neg a = \begin{cases} f, & \text{if } a = t \\ t, & \text{if } a = f \\ a, & \text{otherwise} \end{cases}$



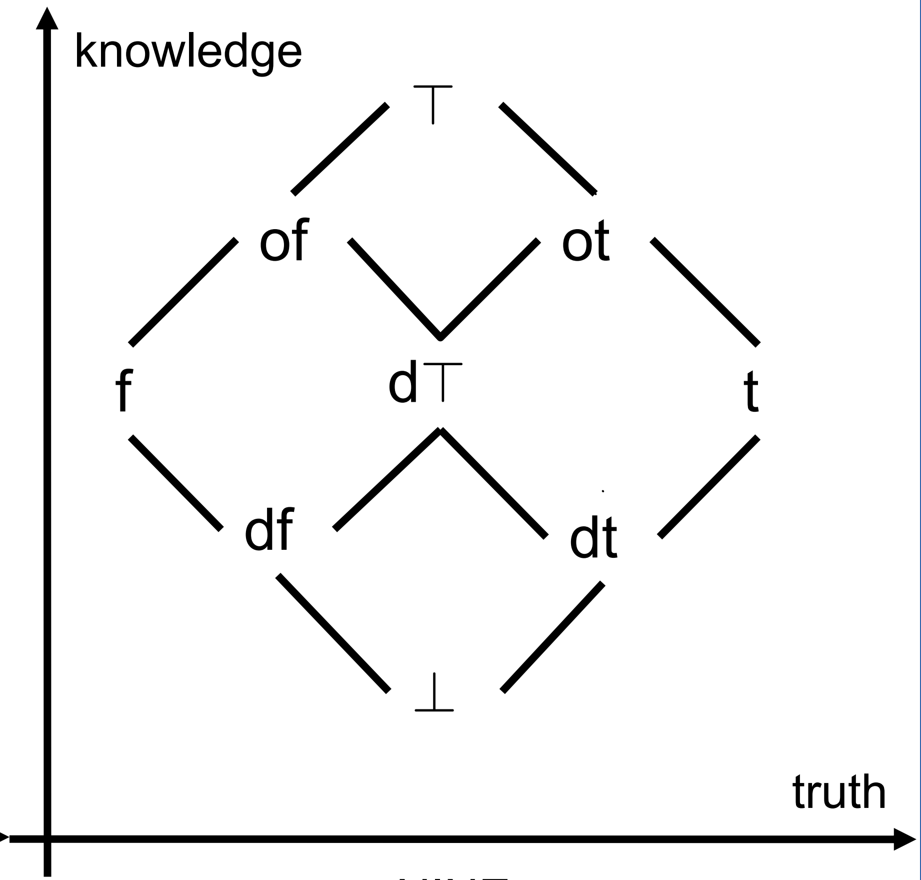
Default f: closed world, default ⊥: open world

Other bilattices

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



SEVEN



NINE

Approach

Generate logical bilattice based on trust order

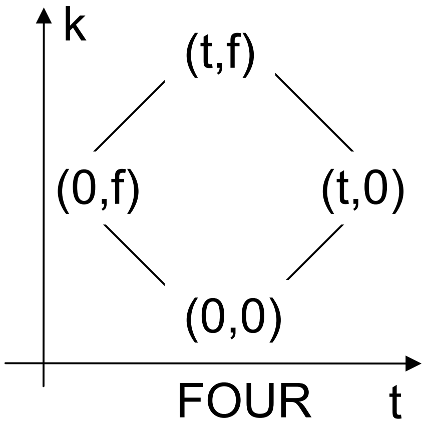
Lukasiewicz:

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

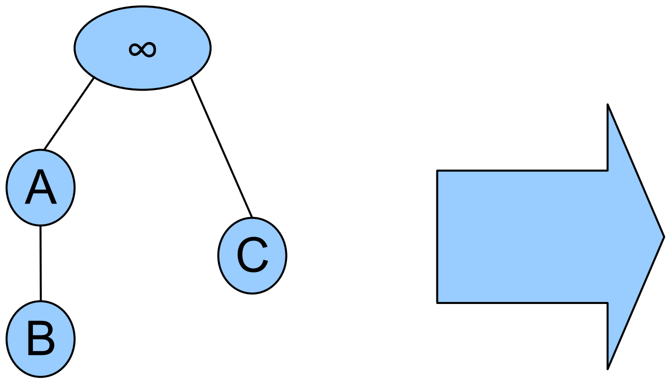
- $(a,b) \leq_k (x,y)$ iff $a \leq_{L_1} x \wedge b \leq_{L_2} y$
- $(a,b) \leq_t (x,y)$ iff $a \leq_{L_1} x \wedge y \leq_{L_2} b$

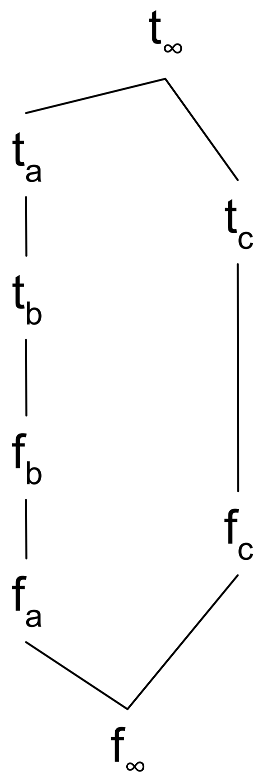
e.g. FOUR = (0,t) × (0,f):



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

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

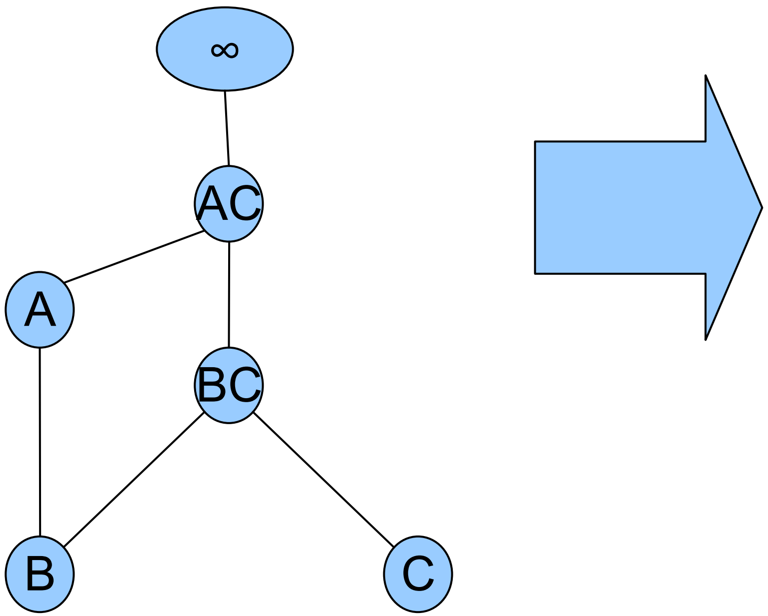


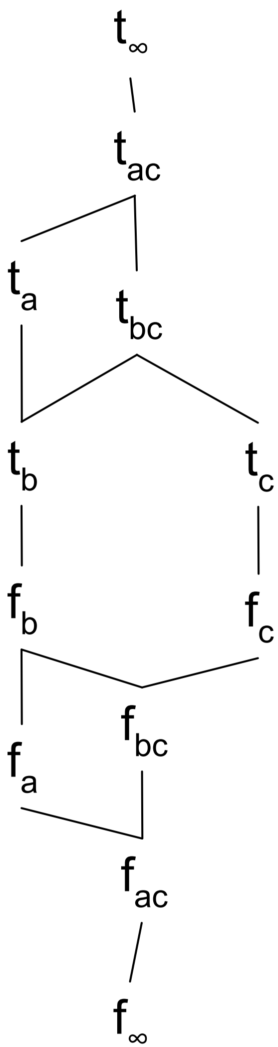


Augmented Trust Order

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

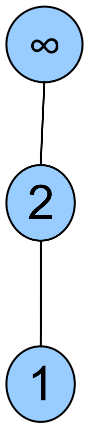
$$L_1 = L_2 = \{(f_i,t_i) \mid i \in S\} \cup$$
$$\{(t_i,t_j) \mid (i,j) \in T\} \cup$$
$$\{(f_i,f_j) \mid (j,i) \in T\}$$

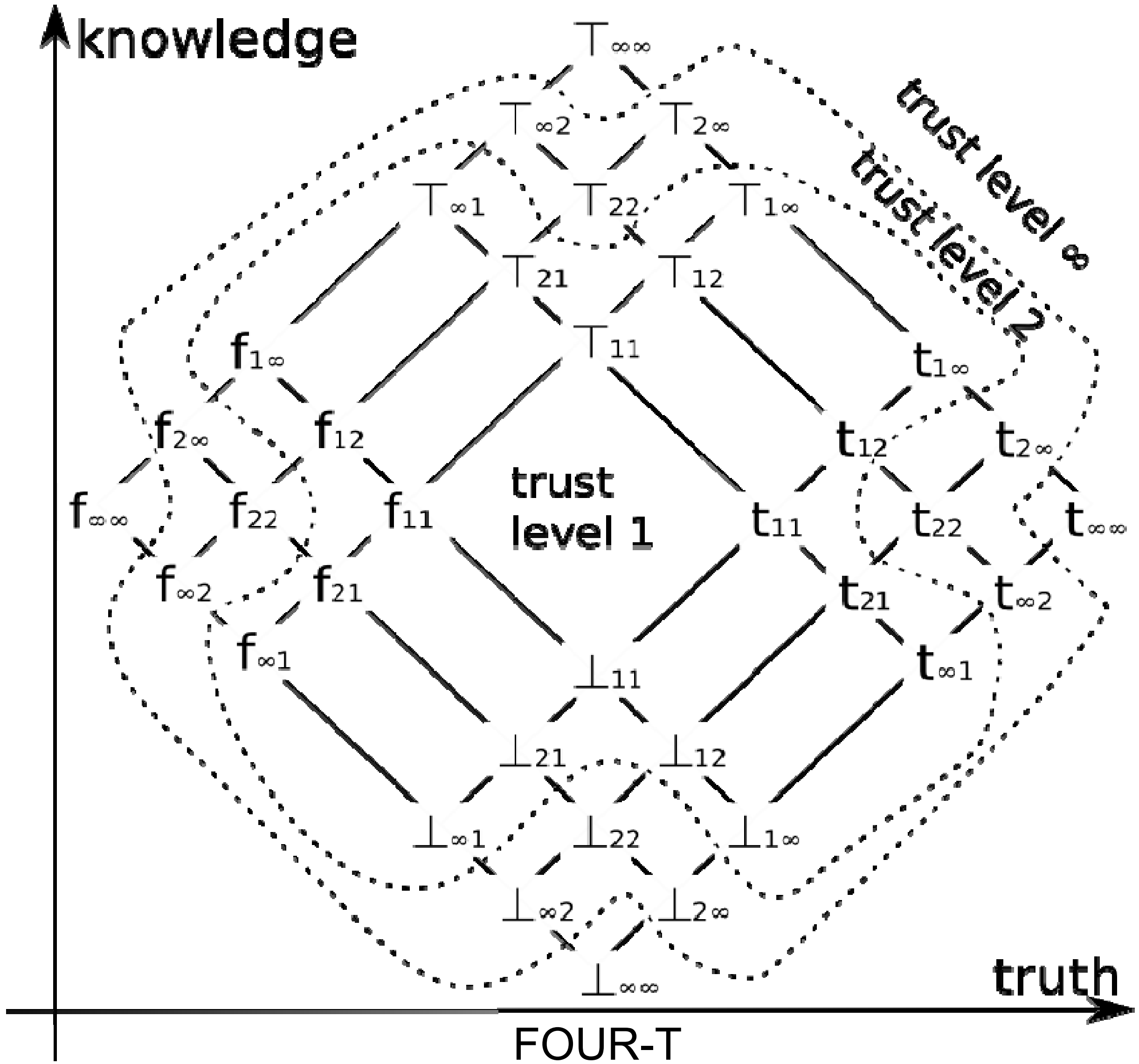




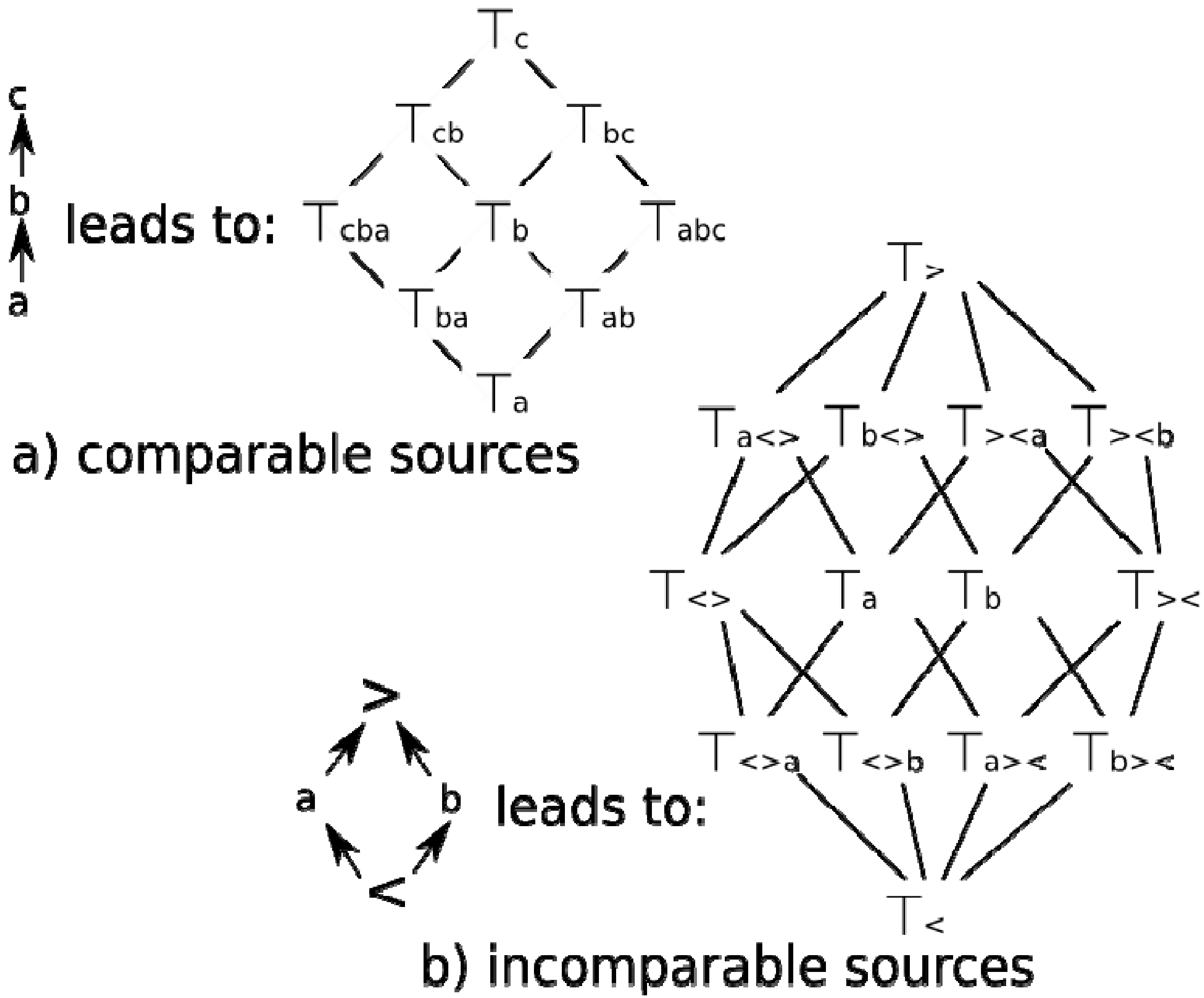
Use trust order
to derive a logical
bilattice.

Example for
comparable
information sources:





FOUR-T


c
b
a
leads to:
$\top_{c}$
$\top_{cb}$ $\top_{bc}$
$\top_{cba}$ $\top_{b}$ $\top_{abc}$
$\top_{ba}$ $\top_{ab}$
$\top_{a}$
a) comparable sources
>
a b
<
leads to:
$\top_{>}$
$\top_{a<>}$ $\top_{b<>}$ $\top_{><a}$ $\top_{><b}$
$\top_{<>}$ $\top_{a}$ $\top_{b}$ $\top_{><}$
$\top_{<>a}$ $\top_{<>b}$ $\top_{a><}$ $\top_{b><}$
$\top_{<}$
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

$(C_1 \sqcap C_2)^I = C_1 \cap C_2$

$(\exists R.C)^I = \{ x: (x,y) \in R^I \wedge y \in C^I\}$

$(C_1 \sqcap C_2)^I(x) = C_1{}^I(x) \wedge C_2{}^I(x)$

$(\exists R.C)^I(x) = \bigvee_{y \in \Delta^I} R^I(x,y) \wedge C^I(y)$

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

Satisfiability wrt. Truth value u: There exists a model, which assigns a truth value $>_t$ u to all axioms

SROIQ on Bilattices (1)

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

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

$$(C_1 \sqcap C_2)^{\mathcal{I}}(x) = C_1^{\mathcal{I}}(x) \dot{\wedge} C_2^{\mathcal{I}}(x)$$

$$(C_1 \sqcup C_2)^{\mathcal{I}}(x) = C_1^{\mathcal{I}}(x) \dot{\vee} C_2^{\mathcal{I}}(x)$$

$$(\neg C)^{\mathcal{I}}(x) = \dot{\neg} C^{\mathcal{I}}(x)$$

$$(S^-)^{\mathcal{I}}(x, y) = S^{\mathcal{I}}(y, x)$$

$$(\forall R.C)^{\mathcal{I}}(x) = \dot{\bigwedge}_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, y) \dot{\rightarrow} C^{\mathcal{I}}(y)$$

$$(\exists R.C)^{\mathcal{I}}(x) = \dot{\bigvee}_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, y) \dot{\wedge} C^{\mathcal{I}}(y)$$

$$(\exists R.\mathrm{Self})^{\mathcal{I}}(x) = R^{\mathcal{I}}(x, x)$$

$$(\geq nS)^{\mathcal{I}}(x) = \dot{\bigvee}_{\{y_1, \ldots, y_m\} \subseteq \Delta^{\mathcal{I}}, m \geq n} \dot{\bigwedge}_{i=1}^{n} S^{\mathcal{I}}(x, y_i)$$

$$(\leq nS)^{\mathcal{I}}(x) = \dot{\neg} \dot{\bigvee}_{\{y_1, \ldots, y_{n+1}\} \subseteq \Delta^{\mathcal{I}}} \dot{\bigwedge}_{i=1}^{n+1} S^{\mathcal{I}}(x, y_i)$$

$$\{a_1, \ldots, a_n\}^{\mathcal{I}}(x) = \dot{\bigvee}_{i=1}^{n} a_i^{\mathcal{I}} = x$$

SROIQ on Bilattices (1)

$$(R \sqsubseteq S)^{\mathcal{I}} = \dot{\bigwedge}_{x,y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x,y) \dot{\rightarrow} S^{\mathcal{I}}(x,y)$$

$$(R = S)^{\mathcal{I}} = \dot{\bigwedge}_{x,y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x,y) \dot{\leftrightarrow} S^{\mathcal{I}}(x,y)$$

$$(R_1 \circ \ldots \circ R_n \sqsubseteq S)^{\mathcal{I}} = \dot{\bigwedge}_{\langle x_1, x_{n+1} \rangle \in dom(S^{\mathcal{I}})} \dot{\bigvee}_{\{x_2, \ldots, x_n\}} \dot{\bigwedge}_{i=1}^{n} R_i^{\mathcal{I}}(x_i, x_{i+1})$$

$$(Asy(R))^{\mathcal{I}} = \dot{\bigwedge}_{x,y \in \Delta^{\mathcal{I}}} \dot{\neg}(R^{\mathcal{I}}(x,y) \dot{\wedge} R^{\mathcal{I}}(y,x))$$

$$(Ref(R))^{\mathcal{I}} = \dot{\bigwedge}_{x \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x,x)$$

$$(Irr(R))^{\mathcal{I}} = \dot{\bigwedge}_{x \in \Delta^{\mathcal{I}}} \dot{\neg} R^{\mathcal{I}}(x,x)$$

$$(Dis(R,S))^{\mathcal{I}} = \dot{\bigwedge}_{x,y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x,y) \dot{\rightarrow} \dot{\neg} S^{\mathcal{I}}(x,y)$$

$$(C \sqsubseteq D)^{\mathcal{I}} = \dot{\bigwedge}_{x \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(x) \dot{\rightarrow} D^{\mathcal{I}}(x)$$

$$(a : C)^{\mathcal{I}} = C^{\mathcal{I}}(a^{\mathcal{I}})$$

$$((a,b) : R)^{\mathcal{I}} = R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}})$$

$$a \approx b = a^{\mathcal{I}} = b^{\mathcal{I}}$$

$$a \not\approx b = a^{\mathcal{I}} \neq b^{\mathcal{I}}.$$

Application: Inconsistency Resolution

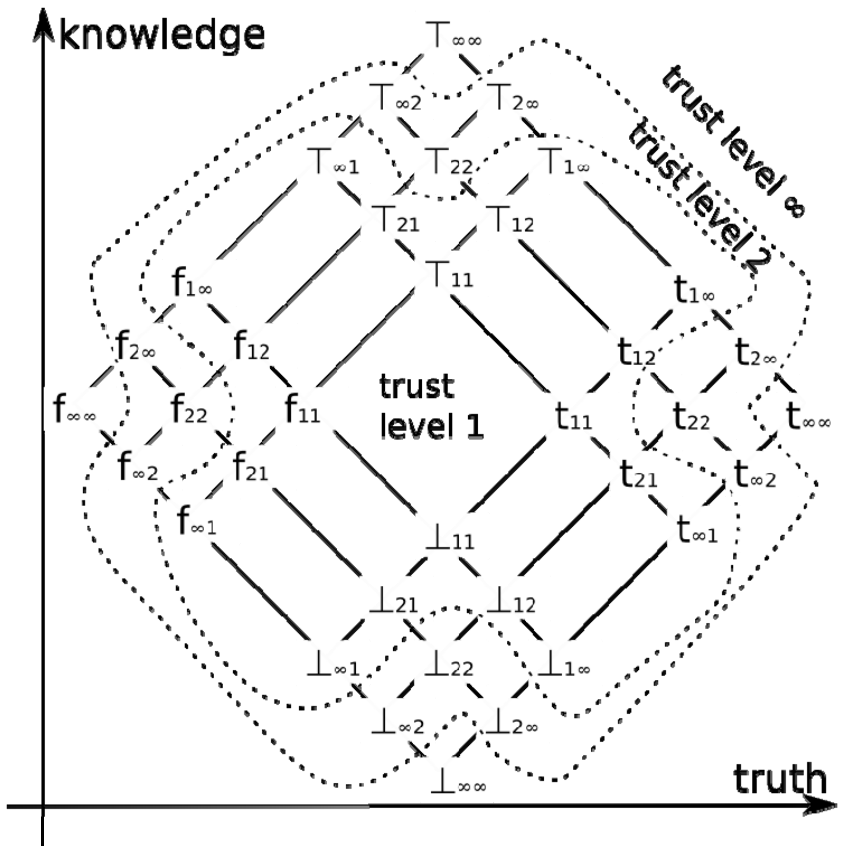
Reasons for Inconsistencies:

$tv(a) = t_x$: $a \leftarrow A$

$tv(a) = f_y$: $a \leftarrow B$

$f_x \wedge t_y = \top_{xy}$ (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 t_{SO} SPIEGEL ONLINE
qos:GoodAction t_{SO}

WELT ONLINE olga:GoodActor f_{W}
qos:GoodAction t_{W}

Mail Online olga:GoodActor f_{M}
qos:GoodAction f_{M}

SROIQ-T

olga:GoodAction → $\top_{W,SO}$

qos:GoodAction → $\top_{M,SO,W} = f_M \oplus t_{SO,W}$

Minimally and maximally trusted source contributing to the inconsistency

MUPS for qos:GoodAction

qos:GoodAction → $t_{SO,W}$

~~qos:GoodAction → f_M~~

Drop minimally trusted axioms

Not possible for 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!