Distributive Lattice Ordered Ontologies

Hans Bruun*

Mai Gehrke†

Jørgen Fischer Nilsson*

* Technical University of Denmark

† Radboud Universiteit

Introduction

- Objective:

  Knowledge representation language

  aimed at uniting

  ontological classification
  and
  relational and object-oriented database representations

- Approach:

  Finite distributive lattices with additional operations and relational semantics

Enriched databases

Creating the enriching structure:
Input: Specification of knowledge domain in terms of generators and relations;
Output: (The dual space of) a particular algebra which solves the generators and relations problem.

Searching enriched databases:
Given the solution from above and a database/set in which each entry is described by a term (pure conjunctions) of the pertinent absolutely free algebra (e.g. lattice type or lattice-with-additional-operations type)
An implementation of a querry mechanism in which one can ask for any term from the absolutely free algebra, and one gets out the information attached to the corresponding term in the solution above.

Specification of ontologies

$\mathcal{O} = (C, A, \Pi)$ Ontological Framework (OF) where

C – A finite set of basic concept names;

A – A finite set of attribute operator symbols;

Π – A finite set of terminological axioms. The elements of Π are pairs (s, t) of DLA terms in the basic concepts.

Generators and relations problem

Solutions of an ontological framework

A solution of $\mathcal{O} = (C, A, \Pi)$ is any quotient

$$h : F_{DLA}(C) \twoheadrightarrow D$$

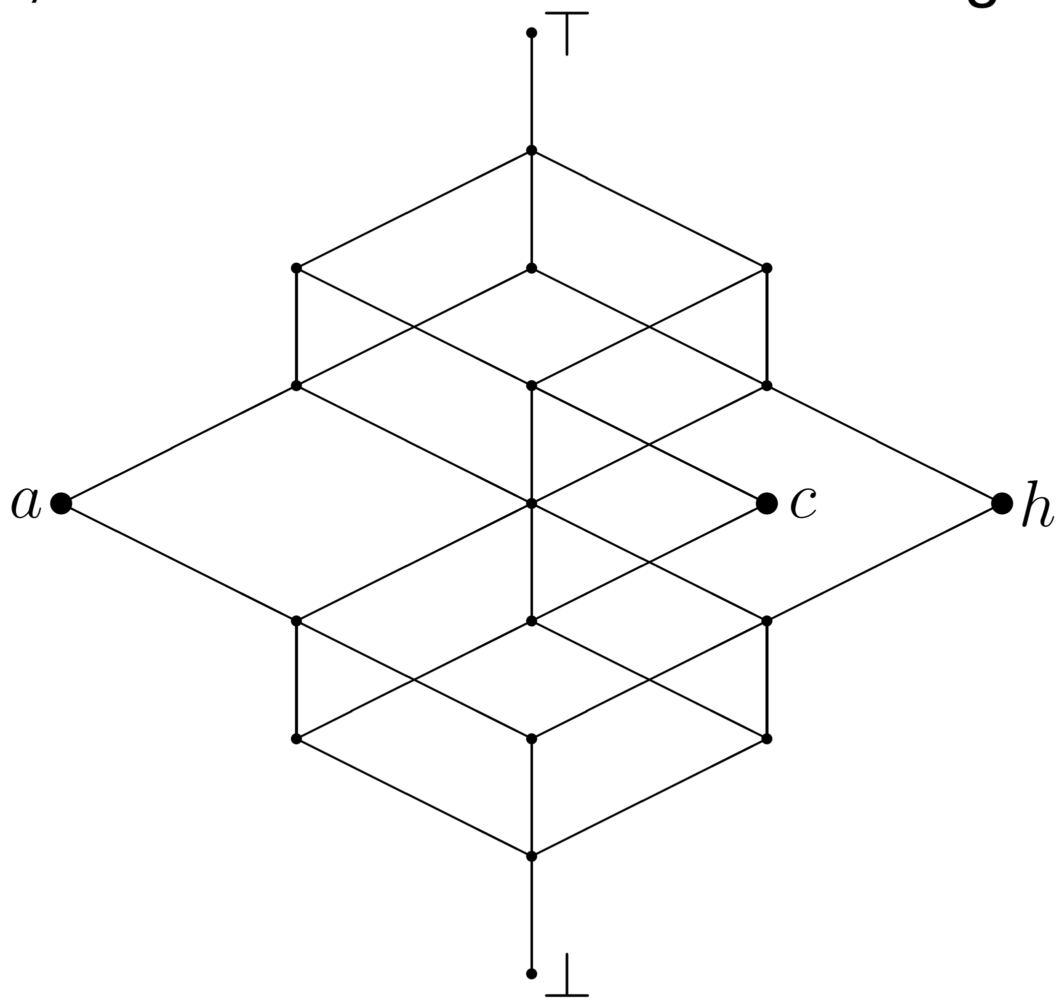
such that

$$\forall\, (r, s) \in \Pi \qquad \text{we have} \qquad h(r) = h(s).$$

Here $F_{DLA}(C)$ is the free algebra in the appropriately defined variety of bounded distributive lattice with attribute operators generated by C.

Example

Let $C = \{h, a, c\}$ where we think of the three concepts as human, adult, and child. With $A = \Pi = \emptyset$ we get:

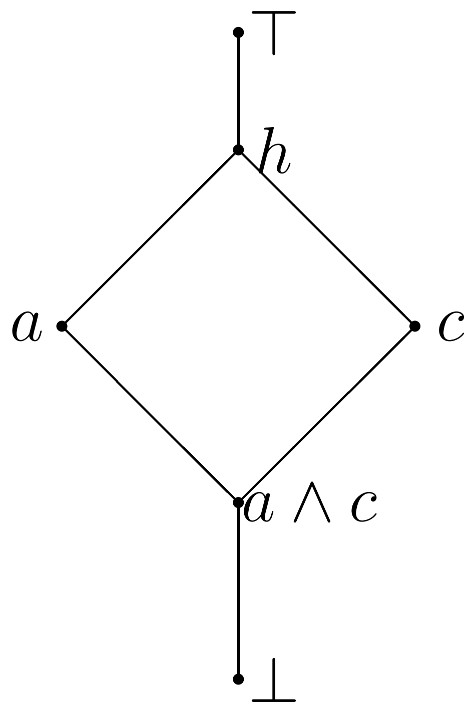


Example - continued

However if we want to identify human with the disjunction of adult and child, we take

$$\Pi \; = \; \{(h, a \vee c)\}.$$

Then we get:



Extending Lattices with Attribution

Axioms for attribution:

$$
\begin{aligned}
a(x \vee y) &= a(x) \vee a(y) \\
a(x \wedge y) &= a(x) \wedge a(y) \\
a(\bot) &= \bot.
\end{aligned}
$$

- The existense results I will talk about all work for most axioms, not just these. Implementation depends on 'incremental' description of free algebras over DL.
- Extension with attributes give rise to nested attribution terms a(t) but one basic intended application is the projections of tuples in a relational database.

The database perspective

- Data base records can be achieved as conjunction (lattice meet) of attributions a(c).
- Database relations can then be achieved as disjunctions (lattice join) of data base records.
- In the algebraic representation database natural join $\bowtie$ is achieved as (a special case of) meet.
- Data base union $\cup$ is achieved as lattice join $\vee$.
- Data base selection is achieved with $\wedge a(c)$ selecting tuples with value c on attribute a (a special case of natural join).

Solutions

From universal algebra we always have a 'first' solution of an OF $\mathcal{O} = (C, A, \Pi)$:

$$h_{\mathcal{O}} : F_{DLA}(C) \twoheadrightarrow F_{\mathcal{O}}$$

where $F_{\mathcal{O}} = F_{DLA}(C)/\Theta(\Pi)$.

The solutions of $\mathcal{O}$ correspond to the further quotients of $F_{\mathcal{O}}$:

$$F_{DLA}(C) \twoheadrightarrow F_{\mathcal{O}} \twoheadrightarrow D$$

Duality

A powerful computational tool

and

a correspondence between language and data.

- In the database model, the conjunctions $id(1) \wedge name(Mai) \wedge mother(Irene) \wedge \ldots$ are the actual data points. In the lattice structure these are the atoms (or more generally the join irreducible elements).
- in the ontology $a(c)$ is the concept of having c in role/as attribute a, whereas the dual is a function f_a by which a datapoint p is sent to its a^{th} coordinate.

Correspondence language vs. data

Ontology	(enriched) dataset
D lattice	$J(D)$ join irreducibles
$\mathcal{D}(P)$ down-set lattice	P poset
$a(c)$= concept of having a-attribute c	$f_a(p)$= the a-attribute of p

Computational tool

Solutions of $\mathcal{O} = (C, A, \Pi)$:

$$q_\theta : F_{DLA}(C) \twoheadrightarrow D$$

Dually:

$$P \subseteq J(F_{DLA}(C)).$$

$$J(F_{DL}(C)) = P(C)$$

Power set of C ordered by dual inclusion or pure conjunctions over C.

$$f_a : c \wedge a(c) \wedge a(d) \wedge a^2(d) \mapsto c \wedge d \wedge a(d)$$

partial order preserving continuous function with clopen up-set domain.

Universal solution in attribute-free case

[F. Oles; Thr.C.S. 2000]

Universal solution of $\mathcal{O} = (C, \Pi)$:

$$F_{\mathcal{O}} \;=\; \mathcal{D}(P_{\mathcal{O}})$$

where

$$P_{\mathcal{O}} \;=\; P \backslash (\bigcup_{(r,s)\in\Pi} Q(r,s))$$

where $Q(r,s)$ is the set of points in $J(F_{DL}(C)) \;=\; P(C)$ disallowed by (r,s).

But this is still large!

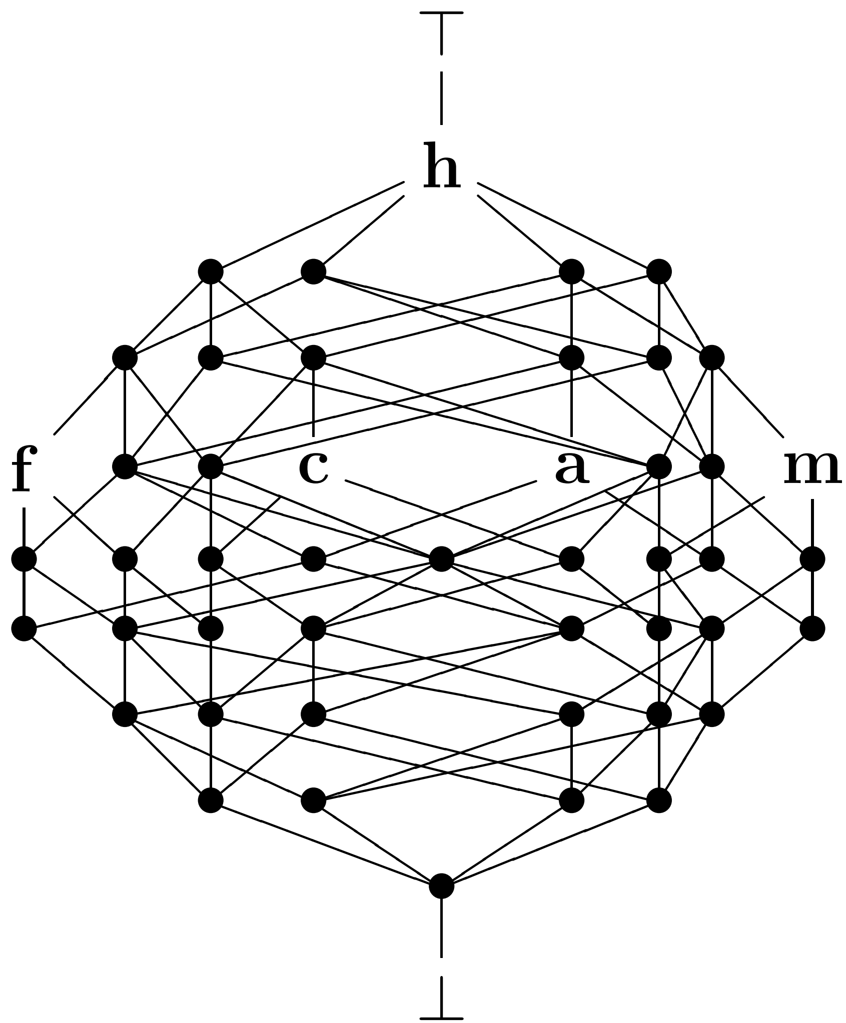
Examples

Example 1:
Let $C = \{h, a, c, m, f\}$ (human, adult, child, male, female)
and $\Pi = \{(h, a \vee c), (h, m \vee f)\}$.
The free DL on five generators has over 7000 elements,
and the solution has 49 (see next slide).

Example 2:
When $A \neq \emptyset$ then typically the universal solution is infinite.

Examples - continued

In example 1 above we get:



The element in the center corresponds to the term

$$(a \wedge c \wedge f) \vee (a \wedge c \wedge m) \vee (c \wedge f \wedge m) \vee (a \wedge f \wedge m)$$

Example 1 - continued

The consequence of $\Pi = \{(h, a \vee c), (h, m \vee f)\}$ for the concept h is that its decomposition into join irreducibles is

$$h \;=\; ham \;\vee\; haf \;\vee\; hcm \;\vee\; hcf$$

The join irreducibles such as $acm,\; acf,\; amf,\; cmf$ have no impact on h.

Specification of a knowledge base

$\mathcal{B} = (C, A, \Pi, I)$ Knowledge Base (KB) where

(C, A, Π) – An ontological framework;

I – A finite set of DLA terms in the basic concepts.

The idea is that the terms in I are the ones that actual data is attached to in the data base, or more generally the ones we want the ontology to classify.

Solutions of a knowledge base

A solution of $\mathcal{B} = (C, A, \Pi, I)$ is any solution of $\mathcal{O} = (C, A, \Pi)$

$$F_{DLA}(C) \stackrel{h_{\mathcal{O}}}{\twoheadrightarrow} F_{\mathcal{O}} \stackrel{h}{\twoheadrightarrow} D$$

such that

$$\forall\, t \in I \qquad \overline{h}^{\flat}(\overline{h}(h_{\mathcal{O}}(t))) \;=\; h_{\mathcal{O}}(t)$$

where $\overline{h} : F_{\mathcal{O}}^{\sigma} \twoheadrightarrow D^{\sigma}$ is the canonical extension of h and $\overline{h}^{\flat}$ is its lower adjoint.

$$F_{\mathcal{O}}^{\sigma} \stackrel{\overline{h}}{\twoheadrightarrow} D^{\sigma} \stackrel{\overline{h}^{\flat}}{\hookrightarrow} F_{\mathcal{O}}^{\sigma}$$

Terminal solution of a knowledge base

Theorem: Under very weak conditions on the type of DL-ordered algebra, KBs over any equational class $\mathcal{V}$ have terminal solutions, that is, solutions

$$F_{\mathcal{V}}(C) \overset{h_{\mathcal{O}}}{\twoheadrightarrow} F_{\mathcal{O}} \overset{h_{\mathcal{B}}}{\twoheadrightarrow} D_{\mathcal{B}}$$

such that any other solution $h : F_{\mathcal{O}} \twoheadrightarrow D$ factors through:

$$\begin{array}{ccc} F_{\mathcal{O}} & \overset{h}{\twoheadrightarrow} & D \\ {\scriptstyle h_{\mathcal{B}}}\downarrow & \swarrow & \\ D_{\mathcal{B}} & & \end{array}$$

Example

$$\mathcal{B} = (C, A, \Pi, I)$$

where

$$C = \{h, a, c, m, f\}, \quad \Pi = \{(h, a \vee c), (h, m \vee f)\}, \quad I = \{h\}$$

Then

$$D_{\mathcal{B}} = \mathcal{D}(P_{\mathcal{B}})$$

where $P_{\mathcal{B}}$ is the anti-chain $\{ham,\ haf,\ hcm,\ hcf\}$

$D_{\mathcal{B}}$ is the 16 element Boolean algebra

Finiteness of the terminal solution

If Π does not have a consequence of the form

$$p \leq a(p) \vee q$$

then the terminal solution is finite.

$$\begin{aligned} p &= (p \wedge a(p)) \vee (p \wedge q) \\ a(p) &= (a(p) \wedge a^2(p)) \vee (a(p) \wedge a(q)) \end{aligned}$$

$$p = (p\, a(p)\, a^2(p)) \vee (p\, a(p)\, a(q)) \vee (p \wedge q)$$

$$\vdots$$

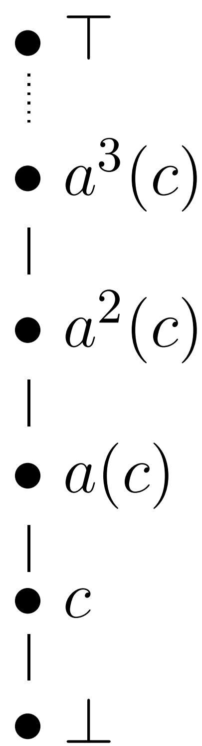
The ontological account of p is infinitely deep and wide.

Example

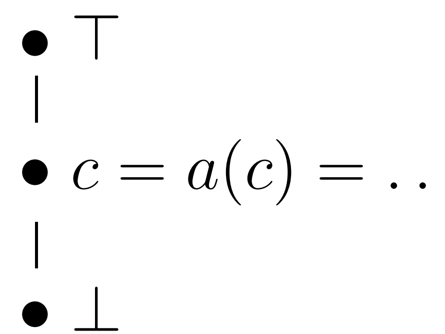
$$C = \{c\},\ A = \{a\},\ \Pi = \{c, c(a(c))\}, I = \{c\}$$

$$c = ca(c)$$

$F_{\mathcal{O}}$



$D_{\mathcal{B}}$



Implementation

- Based on duality approach.
- Yields $(P_{\mathcal{B}}, \leq, \{f_a\}_{a \in A})$ if and only if the terminal solution lies entirely in the finite part of $P(A^*(C))$.
- Querrying over this solution has been implemented in its most rudimentary form.

Further work

- Implementation whenever the terminal solution is finite.
- Implementation for other varieties.
- Expansion of query language and user interface.
- Implementation of various versions of negation:
  - Boolean;
  - pseudocomplement;
  - relative complement;
  - relative pseudocomplement.

Summary

- Distributive lattices extended with attribution operators form a rich and flexible ontology specification language providing equational specifications.
- The framework offers reconstruction of database relations and in particular genralises and simplifies natural join etc.
- The framework is exploited in the ONTOQUERY (see net) project aiming at obtaining content-based access to natural language sources.
- Natural language phrases (NP less determiners) are represented as ground algebraic terms situated in a lattice ontology.
- There is scope for inclusion of the various forms of complementation.