A Logico-Algebraic Framework for Ontologies

ONTOLOG

Jørgen Fischer Nilsson

Informatics and Mathematical Modelling
Technical University of Denmark

1 Introduction

This paper presents a relation-algebraic framework for ontologies taking form of a logic of descriptions termed ONTOLOG. After briefly recalling some constitutional and organisational principles for ontologies, we proceed by choosing algebraic lattices as a "skeleton" language for ontologies taking the concept inclusion relation as basis.

In addition to the well-known lattice diagrams, which facilitate visual presentation of ontologies, the algebraic lattice operations enable compact equational specifications of ontologies with accompanying ontological well-formedness constraints forming what is termed an ontotypology.

The skeleton ontologies are enriched with attribution by way of binary relations (semantic roles) introduced by algebraic composition. The achieved integration of feature structures (structured ontotypes) into ontological lattices with accompanying inheritance constitutes a downwards potentially infinite "generative ontology" enabling dynamic composition and comparison of descriptions. These descriptions are then to serve as representations of noun phrases, e.g., for content-based text retrieval.

Various aspects of the ONTOLOG logical language have been studied previously in [Nilsson 94, Nilsson 99, Larsen & Nilsson 97, Braüner *et al.* 99].

1.1 Some methodological remarks and distinctions

A formal ontology is a logical specification of essential concepts within a domain structured by the relationships identified between the concepts.

According to this definition ontologies differ from thesauri (cf. e.g. [Eco 95] for a historical account) by putting emphasis on the concepts rather than the words denoting the concepts, and furthermore by relying on a formal logical framework for structuring the ontologies.

One can distinguish at least the following fundamental aspects of ontologies:

1. Formal logical representation languages for ontologies
2. Constitutional and organisational principles for ontologies
3. Concrete proposals for general and specific ontologies
4. Supporting system functionalities

These aspects are of course connected: Elaboration of an ontology calls for an appropriate formal framework and involves commitment to some organisational principles.

In the course of presenting the ONTOLOG language in its concrete shape, we also approach and commit ourselves to some constitutional and organisational principles.

2 Concepts and relationships

The basic referents in the ONTOLOG language are concepts and (binary) relationships between concepts. Concepts comprise:

- general concepts (classes), also called universal concepts or just universals.
- individual concepts (individuals or particulars).

Linguistically, general concepts correspond to nouns (including nominalised verbs), whereas individuals correspond to proper nouns, at least in a philosophically naive view.

2.1 Extension versus intension

In the so-called extensional view a concept reduces to the set of entities falling under the concept. For instance, the *extension* of `cat` is the set of all cats, etc. For individual concepts the extension is the single entity comprised by the concept.

However, concepts are not sets since, unlike sets, concepts possess an *intension* understood as all the properties shared by entities falling under the concept. This means that two concepts may be *co-extensional*, i.e. have the same extension, yet being distinct.

Empty concepts are concepts with an empty extension set.

It is important to realise that an ontology is intended to account for the intensional aspects of concepts.

3 Classification and taxonomy

The concepts to be included and modeled in an ontology are placed in a type structure or taxonomy for the target domain. For this reason in ONTOLOG the term ontotype is often used for the concepts selected for representation in the ontology.

4 Conceptual inclusion

The basis for the classification is here as usual the conceptual inclusion relationship *isa*:

$$c' \xrightarrow{isa} c''$$

expressing that c' is a sub-concept of c'', and conversely that c'' is a super-concept of c'. In other words, c' is a specialization of c'' and c'' is conversely a generalisation of c'.

In ONTOLOG notation is written

$$c' < c''$$

or in case that c' and c'' are not necessarily distinct: $c' \leq c''$.

Graphically in diagram notation for the strict inclusion $c' < c''$ is used

$$\begin{array}{c} c'' \\ \uparrow \\ c' \end{array}$$

We follow the convention of omitting the *isa* name tags on inclusion arcs. Moreover is adhered to the common principle of letting inclusion arcs be upwards directed[1]. Thus arrow heads can be disposed of in graph diagrams.

Individual concepts have no subconcepts, except for a distinguished degenerate concept, the empty `null` concept, to be introduced below for formal reasons.

In diagrams a concept c is situated between its immediate superconcepts and subconcepts:

$$\begin{array}{c} \nwarrow\uparrow\nearrow \\ c \\ \nearrow\uparrow\nwarrow \end{array}$$

In the below figures a diamond shaped frame is introduced, which is intended to associate to the lattice structure inherent in ONTOLOG.

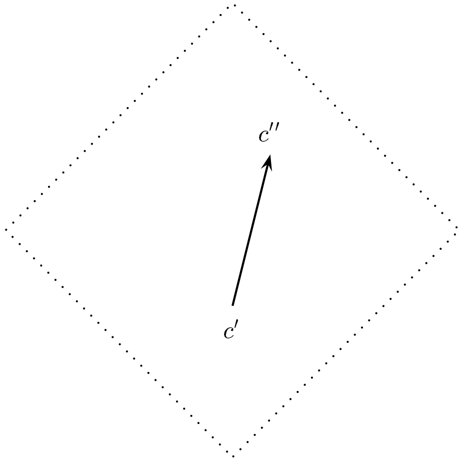

Figure 1: Sub- and super-concept

From sect. 12. we address conceptual relationships (other than *isa*), which may exist between considered ontotypes.

4.1 A note on partof-relationships

The inclusion relationship should not be confused with the part-of relationship[2]. For instance, France is a country, but Provence is a part of France. Therefore

[1] This is based on the presumption that the inclusion relationship is an ordering relationship, so that there are no proper, cyclic, strict *isa* relationships.

[2] These two relationships seem to be identified in the so-called Mereology, see e.g. [Simons 87, Sowa 00].

we consider in a later section a separate part-of relation on a par with other general purpose relationships.

5 Hierarchical classification

We now consider ways in which the inclusion relationship may serve to establish classifications. In the course of this various operations are introduced on concepts. These operations are often more useful than the inclusion relationship itself when specifying ontologies.

The simplest form of classification is the hierarchical one taking form of trees typified by the pattern

$$\begin{array}{ccc} & c & \\ \nearrow & & \nwarrow \\ c_1 & & c_2 \end{array}$$

conforming with the specification $c_1 \ < \ c$ and $c_2 \ < \ c$.

6 Operations on ontotypes

The hierarchy above can also be specified equationally as

$$c = c_1 + c_2$$

using the conceptual sum, $+$, to be described further below. This equation leaves open whether c_1 and c_2 are disjoint. Equational specifications admit often very compact specifications of complex taxonomies as we shall see.

This conceptual structure can be depicted in ONTOLOG diagrams as

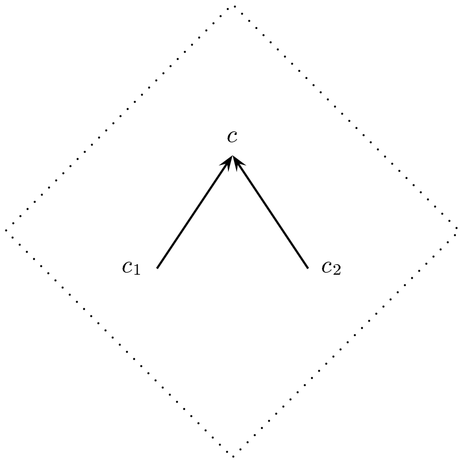

Figure 2: Concept hierarchy

In an extensional view of concepts the sum $+$ reduces to set union.

If we explicitly want to indicate that there may be other contributions to c besides c_1 and c_2 this can be achieved by introducing an anonymous ontotype (node).

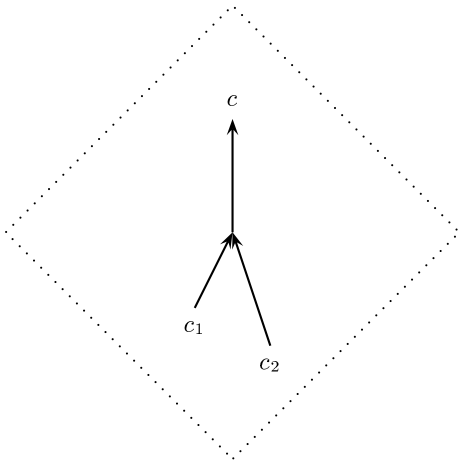

Figure 3: Open Concept hierarchy

7 Trans-hierarchical structures

Often in ontologies there are overlapping subclasses typified by the pattern

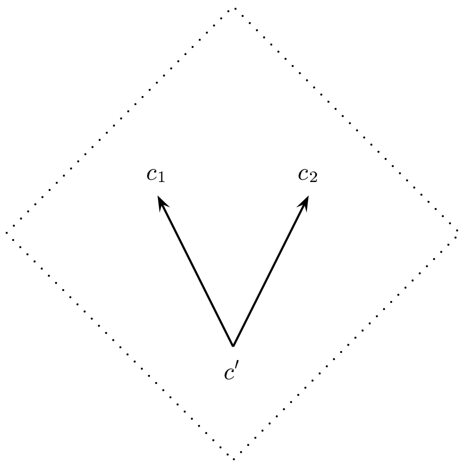

Figure 4: Crux'ed ontotype concept

with pairs of overlapping ontotypes c_1 and c_2 as in the diagram.

Dually to the above equational specification this case can be specified as

$$c' = c_1 \times c_2$$

where the operator $\times$ is the conceptual product, called crux. Extensionally this operator is understood as set intersection.

Combining these two classification patterns yields common ontological patterns as exemplified with

$$\begin{tikzcd}
 & \mathsf{human} & \\
\mathsf{child} \arrow[ur] & & \mathsf{male} \arrow[ul] \\
 & \mathsf{boy} \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

This example conforms with the basic lattice structure diagram in fig. 5 below, corresponding to the equations

$$c_{sup} = c_1 + c_2$$
$$c_{inf} = c_1 \times c_2$$

where names c_{sup} and c_{inf} are introduced in order to associate to the notions of supremum and infimum in lattices, cf. [Davey & Priestley 90].

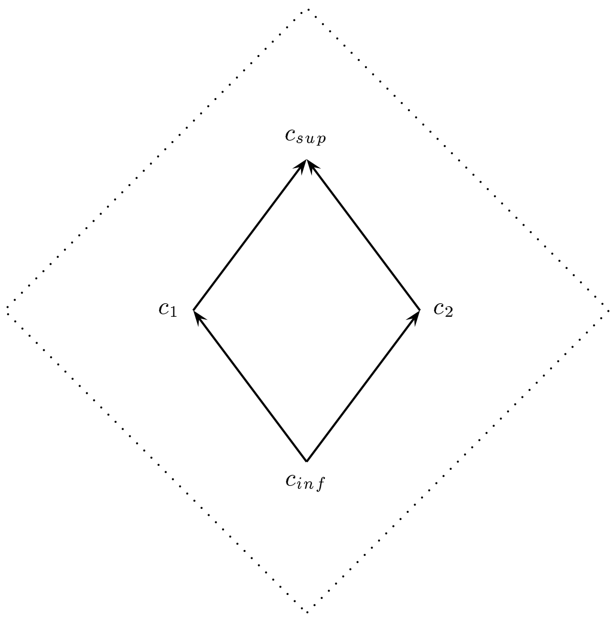

Figure 5: Ontotypes with sum and crux

Further examples are

$$\begin{array}{ccccc} & & \textsf{furniture} & & \\ & \nearrow & & \nwarrow & \\ \textsf{sofa} & & & & \textsf{bed} \\ & \nwarrow & & \nearrow & \\ & & \textsf{sofabed} & & \end{array}$$

$$\begin{array}{ccccc} & & \textsf{meal} & & \\ & \nearrow & & \nwarrow & \\ \textsf{breakfast} & & & & \textsf{lunch} \\ & \nwarrow & & \nearrow & \\ & & \textsf{brunch} & & \end{array}$$

7.1 Conceptual disjointness

Disjointness of a pair of concepts can be specified explicitly by stating that their single common subconcept is the distinguished empty null concept as in

$$\begin{tikzcd} & \mathsf{pet} & \\ \mathsf{cat} \arrow[ur] & & \mathsf{dog} \arrow[ul] \\ & \mathsf{null} \arrow[ul] \arrow[ur] & \end{tikzcd}$$

It turns out to be feasible to rely on a default convention according to which two concepts are considered disjoint unless there is explicitly given a proper (i.e. non-null) common subconcept in the ontology. This conforms with the convention in hierarchical classifications.

Disjointness holds in particular for pairs of individuals (except for synonyms).

8 Lattice structures and diagrams

The adoption of lattices as a formal representation language for ontologies is a commitment to the following existential presumptions:

1. For any pair of concepts c_1 and c_2 there exists a least superior concept, c_{sup}, such that c_{sup} is a subconcept of or equal to any other superior concept of c_1 and c_2.
2. Dually, for any pair of concepts c_1 and c_2 there exists a largest inferior concept, c_{inf}, such that c_{inf} is a superconcept of or equal to any other inferior concept of c_1 and c_2.

In our opinion these existence requirements can always be fulfilled by ontologists by formal stipulation of appropriate intermediate anonymous concepts. The case of unrelated concepts may be coped with formally by adoption of top and bottom of the entire lattice ontology as supremum and infimum.

9 Structure of ontologies

The skeleton ontologies formalised as finite lattices form a map of the ontological landscape, which is roughly structured in two dimensions.

9.1 Stratification of ontology

Ontologies may be structured by horisontal layers and orthogonally into vertical divisions into regions.

The horisontal stratification distinguishes an upper section of general ontotypes, a medium level, and a lower level of more specialised concepts.

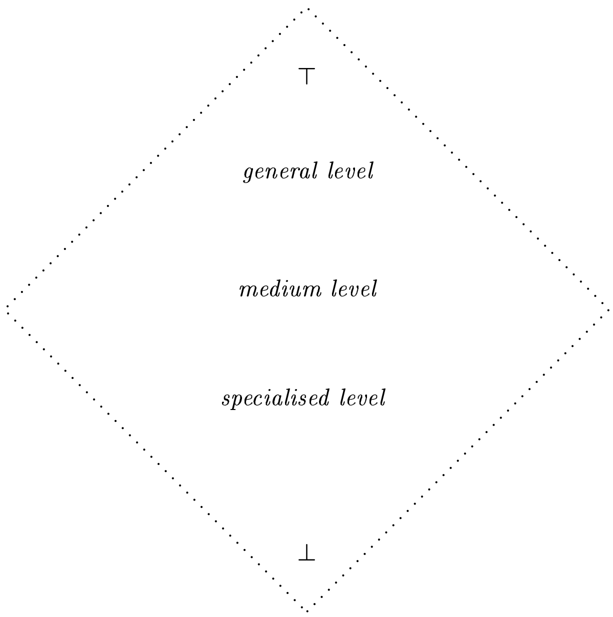

Figure 6: Stratification

This figure also shows the lattice boundary values $\top$ (top) and $\perp$ (bottom). For the latter term in this context is used the symbol null.

9.2 Regionalisation of ontology

Orthogonally to the stratification the skeleton ontology may be divided into regions corresponding to the classification into general major categories such as, say, Time, Place, Occurrrent, Property.

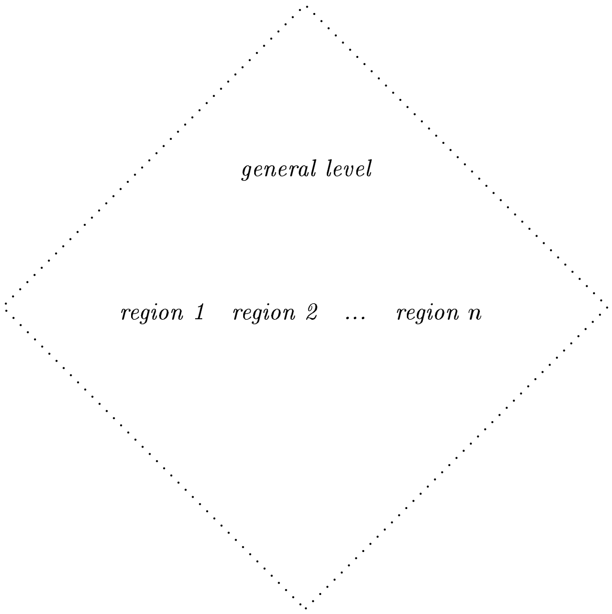

Figure 7: Regionalisation

The ontotypes in different regions are not necessarily disjoint.

Individuals reside in the lower triangle of the skeleton ontology.

10 Lattices as algebras

Above we have considered the concept comparison relation $\leq$ (conforming with concept inclusion) and the two closed operations of sum and crux on ontotypes. This section re-introduces these operations more formally algebraically. The algebraic formalisation admits computational comparison and reasoning with the formal representation of the considered concepts as ontotypes.

10.1 Algebraic terms for concept descriptions

A (concrete) algebra is a collection of elements of a universe and some closed operations (everywhere defined functions) on these elements. The notion of algebra is well-known from arithmetic with closed operations such as sum and product on numbers.

We consider concept algebras where the elements are understood to be (descriptions of) concepts called ontotypes.

The algebra introduces expressions called terms, φ and ψ, being
– constants (concept names, ontotypes c_i),
– variables (for use in algebraic laws), and
– compound terms comprising an operator with its accompanying sub-terms.

The identity of two ontotypes may be expressed with an equation $\varphi = \psi$.

10.2 Operations on concept descriptions

On concept terms φ and ψ are introduced the following closed operations

- The conceptual *sum* of φ and ψ written $(\varphi + \psi)$ understood as the concept of being either φ or ψ (or both).

  Thus sum is related to logical disjunction as well as to set union.

- The conceptual product of φ and ψ is here called *crux* in order to avoid confusion with Cartesian product. It is written $(\varphi \times \psi)$, and is understood as the concept of being φ as well as ψ.

  Thus crux is related to logical conjunction as well as to set intersection.

For $(\varphi \times \psi)$ is often used just $(\varphi\ \psi)$. Often the parentheses are omitted in compound terms.

As an example we can form concept descriptions (compound terms)

$$\mathsf{dog} + \mathsf{cat} \quad \text{and} \quad \mathsf{dog} \times \mathsf{cat}$$

The latter compound ontotype is equal to the empty concept null (lattice bottom $\perp$).

The algebraic axioms formalising this algebra are given in the appendix. The so-called distributive lattices are used in order to ensure an underlying set-theoretical interpretation.

The axioms are available as rewriting rules. As a simple example the term dog + cat is identical to cat + dog by virtue of the commutative law.

This algebra is a so-called pre-Boolean algebra: By addition of a complement operation (cf. negation) may be obtained the well-known notion of Boolean Algebra. However, we consider the complementation of concepts to be alien to ontologies.

10.3 Sub-ontologies and abstracted ontologies

The algebraic framework admits various notions for structuring and abstracting ontologies.

- The so-called sub-lattices are selected parts of lattice diagrams closed under the considered operations.
- The so-called homomorphisms are structure-preserving mappings of lattices into lattices, which may "shrink" part of a diagram into a single point, thereby providing a simplified image of an ontology.

It should be noticed that anonymous (non-lexicalised) points may be brought into existence by the operations of the algebra. Such anonymous points may be omitted (left implicit) in diagrams.

11 Equational specifications of ontologies

An important aspect of ONTOLOG is that it by virtue of being an algebra admits equational specifications of ontologies complementary to the visualisation by diagrams.

We have already met tiny ontologies in equational form:

$$\mathsf{pet} = \mathsf{dog} + \mathsf{cat}$$
$$\mathsf{dog} \times \mathsf{cat} = \mathsf{null}$$

To this equational ontology may be added for instance the inequality

$$\mathsf{pet} < \mathsf{animal}$$

The equational form represents a compact alternative to the diagram form of ontologies presented above for larger ontologies. Moreover the algebraic term form enables logical calculation with concepts in ontologies similarly to calculation with numbers in arithmetic, as we shall see.

The equations may be conceived of as a kind of type declarations for ontotypes similar to data types in programming languages.

It should be observed that multiple definitions may be introduced, for instance

$$\mathsf{human} = \mathsf{male} + \mathsf{female}$$
$$\mathsf{human} = \mathsf{child} + \mathsf{adult}$$

with the accompanying

$$\mathsf{male} \times \mathsf{female} = \mathsf{child} \times \mathsf{adult} = \mathsf{null}$$

This ontology generates a quite complicated diagram with 4 (initially anonymous) subconcepts: man, woman, boy, and girl.

12 General relationships in ontologies

So far we have only considered skeleton ontologies with the concept inclusion as the sole structural relation.

Now we enrich these taxonomies with assorted binary relationships between ontotypes, e.g. for the purpose of representing thematic roles.

At the informal level binary relationships r between concepts can be introduced simply as labeled arcs connecting the relata in lattice diagrams as known from semantic nets, viz.

$$c_1 \xrightarrow{r} c_2$$

A key issue is then the formal integration of such ontological relationships into the above introduced logico-algebraic framework.

12.1 Peirce product

Since the logico-algebraic understanding of lattices is to be maintained, binary relationships r are introduced algebraically by means of an operator, namely a distinguished dyadic operator, the the so-called Peirce product, cf. [Brink *et al.* 94], symbolised by colon as in

$$(r : \varphi)$$

which combines a binary relation r with an ontotype term φ to give an ontotype. Thus r relates nodes in the ontology[3].

The Peirce product is typically used as a factor as in

$$c \times (r : c_1)$$

In the basic application in ONTOLOG the relation r in the Peirce product $(r : \varphi)$ is an identifier [4]. In this case $(r : \varphi)$ can be replaced with

[3]The Peirce product conforms with the set explication: $[\![(r : \varphi)]\!] = \{x | \exists y(\langle x, y\rangle \in r \wedge y \in [\![\varphi]\!])\}$ corresponding to the abstraction $\lambda x.(\exists y(r(x, y) \wedge \varphi(y)))$.

[4]In a concluding section we discuss the more general case of compound relational terms for r.

$$r(\varphi)$$

where formally r arises as a unary operator from ontotypes to ontotypes.

In an intuitive understanding the ontotype $c \times r(c_1)$ functions as a feature structure conventionally written

$$c\ [r : c_1]$$

with the attribute value pair $[r : c_1]$ attaching to the node c conforming with the diagram

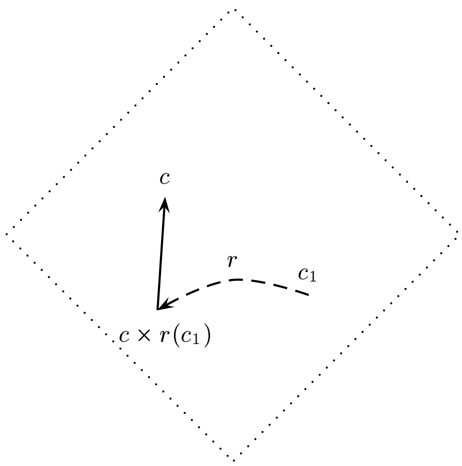

Figure 8: Attachment of relational attribute

Thus r functions dually as a binary relation and as a role name interlocked with the ontological structure.

13 Attribution and feature structures

As indicated above in ONTOLOG *ad hoc* binary relations r are introduced in their simple non-compound form by the term $r(\varphi)$. Since this term functions as an attribute-value pair we write $a(\varphi)$ with a for attribute and often use the term in the form $c \times a(c_1)$.

This is like a feature structure, cf. e.g. [Carpenter 92, Rounds 97],

$$c\ \left[\ a : c_1\ \right]$$

describing those c which possess the a attributed concept value of c_1, and visualised as follows in combined notation of lattice diagram and feature structure

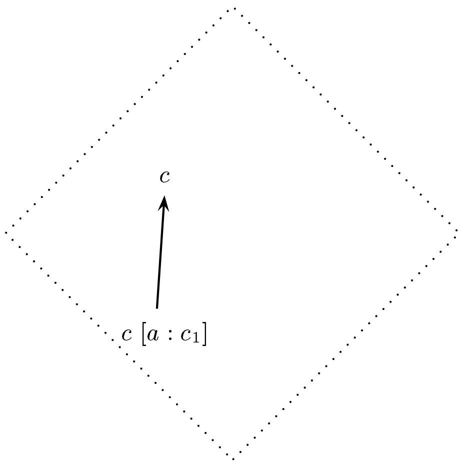

Figure 9: Specialisation by attribution

Examples:

$$\textsf{production}\ \left[\ rst : \textsf{wine}\ \right]$$

specialises production, and similarly

$$\textsf{lack}\ \left[\ wrt : \textsf{vitaminA}\ \right]$$

specialises lack.

14 Axioms for attribution

The following axioms are introduced for each attribute a

$$a(X + Y) = a(X) + a(Y)$$

$$a(\perp) = \perp$$

The last axiom expresses that the null concept in an attribution nullifies the whole feature structure.

In addition, for the so-called functional features there is

$$a(X \times Y) = a(X) \times a(Y)$$

14.1 Monotonicity of attribution

Binary relations do not in general introduce any inclusion order as indicated:

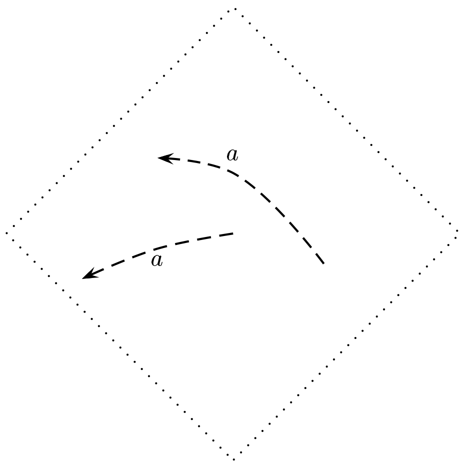

Figure 10: Relations do not impose inclusion order

However, from the above axioms follows *monotonicity of attribution*:

$$X \leq Y \ \rightarrow \ a(X) \leq a(Y)$$

This law is illustrated with

$$\mathsf{lack}\ \left[\ wrt : \mathsf{vitaminA}\ \right]$$

being a specialisation of

$$\mathsf{lack}\ \left[\ wrt : \mathsf{vitamin}\ \right]$$

given that vitaminA specialises vitamin. Here WRT, i.e., “with respect to”, is a general purpose ontorelation, cf. sect. 15.

Similarly production of red wine, $\mathsf{production}\ \left[\ rst : \mathsf{red\_wine}\ \right]$, is a specialisation of

$$\mathsf{production}\ \left[\ rst : \mathsf{wine}\ \right]$$

given that red_wine specialises wine.

The logical property of monotonicity can be visualised as follows:

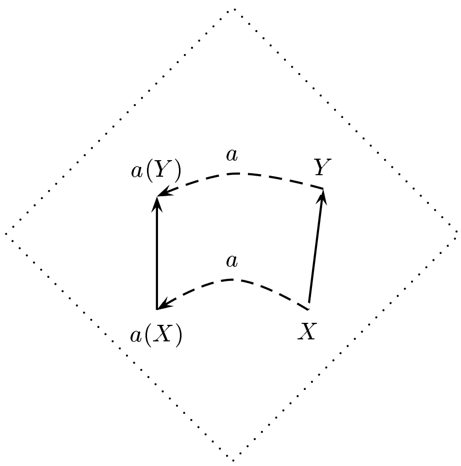

Figure 11: Monotonicity of attribution

14.2 General feature structures

As a straightforward generalisation of the above, multi-entry feature structures (frames)

$$c \begin{bmatrix} a_1 : \varphi_1 \\ ... \\ a_m : \varphi_m \end{bmatrix}$$

are achieved by the ONTOLOG term

$$c \times a_1(\varphi_1) \times a_2(\varphi_2) \times ... \times a_m(\varphi_m)$$

or in shorter form discarding $\times$: $c\ a_1(\varphi_1)\ a_2(\varphi_2)\ ...\ a_m(\varphi_m)$.

Example:

$$\textsf{production} \begin{bmatrix} rst : \textsf{wine} \\ loc : \textsf{France} \end{bmatrix}$$

One observes that the cruxing of additional entries functions as a (further) specialisation of the entire term.

Frames may be nested as in:

$$\mathsf{production}\left[\, rst : \mathsf{wine}\left[\, chr : \mathsf{red} \,\right] \right]$$

conforming with the algebraic term

$$\mathsf{production} \times rst(\mathsf{wine} \times chr(\mathsf{red}))$$

14.3 Inheritance

Attached feature structures are inherited downwards in the lattice as a consequence of the algebraic axioms. Thus there is no auxiliary computational machinery needed to obtain downwards inheritance of attached properties. In case of inheritance from multiple super-concepts this may give rise to conflicting inheritance.

The above attribute axioms are instrumental in the inheritance of properties as indicated with the below lattice diagram fragment:

$$\begin{tikzcd} \varphi \times a(\varphi') & & \psi \times a(\psi') \\ & \varphi \times \psi \times a(\varphi' \times \psi') \arrow[ul] \arrow[ur] & \end{tikzcd}$$

This diagram explains how a joint attribute value $\varphi' \times \psi'$ arises by multiple (possibly conflicting) inheritance from two superior ontotypes sharing the functional attribute a.

The case of disjointness (i.e., $\varphi' \times \psi' = \mathsf{null}$) reduces $\varphi \times \psi \times a(\varphi' \times \psi')$ to the empty description null, which is a neutral element in sums.

15 Thematic roles as ontorelations

Of particular interest in ontologies are universally applicable semantic relations or thematic roles (originating in Fillmore's classical cases), here called ontorelations exemplified in the below open-ended table.

As it appears these roles express temporal, spatial, structural, causal, instrumental, property-ascribing, and other forms of relationships. The present list is not meant to be exhaustive, though it is supposed to cover most roles needed in ONTOLOG. A comprehensive survey of semantic relations is given in [Madsen *et al.*] (present volume), see also [Somers 87] for a linguistic study.

Role	Legend
TMP	temporal aspects (generic role)
LOC	location, position
PRP	purpose, function
WRT	with respect to
CHR	characteristic (property ascription)
CUM	cum (i.e., with accompanying)
BMO	by means of, instrument, via
CBY	caused by
CAU	causes
CMP	comprising, has part
POF	part of
AGT	agent of act or process
PNT	patient of act or process
SRC	source of act or process
RST	result of act or process
DST	destination of moving process
?	?

The TMP role is to be elaborated into specialised temporal roles such as start and duration, etc.

15.1 Algebra for relations

In the present basic form of ONTOLOG the relation r in the Peirce product $(r : \varphi)$ is an identifier. As mentioned, in this case $(r : \varphi)$ can be replaced with $r(\varphi)$ stressing the use of r as a feature or attribute.

In the general use of the Peirce product $(r : \varphi)$ in [Brink *et al.* 94] r may be a compound term in a separate lattice algebra of binary relations.

This means that in the general case there are also lattice operations $+$ and $\times$ available on relations. In this relational algebra there are rules like

$$((r_1 + r_2) : \varphi) = (r_1 : \varphi) + (r_2 : \varphi)$$

As an example the above causal roles CAU and CBY may be understood as each others inverse. Then their relational sum (CAU + CBY) expresses an undirected causal connection.

In addition to the lattice operations on relations, there may be closed relational operations such as composition of relations and inversion of a relation.

15.2 Relational basis

It is our tenet that a multitude of more specialised semantical relations can preferably be re-expressed by means of a limited base of universal roles such as the above list.

For instance a treatment relationship, "a *treats* b", can be re-expressed (appealing to the principles of Davidsonian semantics) as the ontotype

$$\mathsf{treatment}\begin{bmatrix} agt : \mathsf{a} \\ pnt : \mathsf{b} \end{bmatrix}$$

This representation possesses the advantage that more attributes e.g. concerning the temporal aspect can be added (as a term factor in the algebra), thus avoiding variant relations such as "*owned*". Moreover, the introduced sample ontotype can find its proper place in the ontology as a special kind of legal state of affairs close to related concepts such as borrowing situations.

In the same vein the relationship "aspirin is used against pain" can be reformulated as the ontotype

$$\mathsf{treatment}\begin{bmatrix} bmo : \mathsf{aspirin} \\ pnt : \mathsf{pain} \end{bmatrix}$$

which subsumes, say

$$\mathsf{treatment}\begin{bmatrix} bmo : \mathsf{aspirin} \\ pnt : \mathsf{headache} \end{bmatrix}$$

given that headache is a sub-concept of pain in the ontology.

Adhering to this exchange principle *ad hoc* relations can thus be eliminated in favour of a richer structure of ontotypes.

16 Ontotypology

We suggested adoption of a default principle stating that the crux of two ontotypes reduces to null unless stated otherwise in the ontology.

Appealing to this principle an ontology also serves to specify combinability of concepts, thereby effectively imposing type restrictions on the ontotypes.

Such ontological well-formedness restrictions are referred to as the ontotypology. In particular the ontotypology is to state the admissible concept attributions, preferably at higher levels in the ontology so that they are distributed by inheritance.

17 The notion of generative ontology

Although the skeleton ontology is assumed to be a finite lattice, the inclusion of attribute terms makes the ontology "generative", since ever more nodes corresponding to specialised concepts can be formed downwards in the ontology as sketched in the below figure 12.

This means that the lattice is no longer atomic in the lattice-theoretic sense.

Existence of potentially infinite chains in the ontology makes it possible to accommodate ontological representations of an indefinite number of noun phrases.

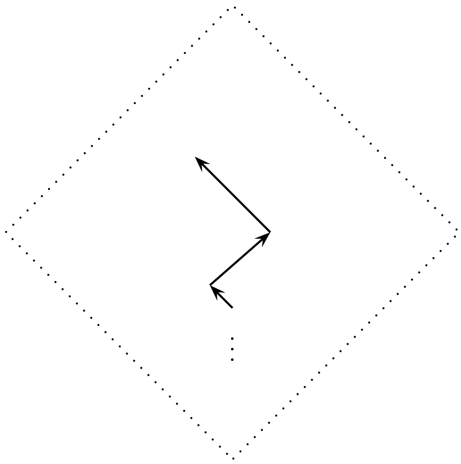

Figure 12: Downwards potentially infinite chain

18 Compositional ontological semantics

Lexical items may be related to the ontology simply by a mapping from nouns and other entries into corresponding ontotype nodes in the ontology. Synonyms are to be mapped into the same node.

Compound nouns are typically mapped into feature structure nodes.

Polysemy may be handled by mapping the lexical entry into an ontological sum node.

As mentioned, not all nodes in the ontology need to be lexicalised.

Let $\langle\!\langle P \rangle\!\rangle$ denote the node (ontotype) associated with the lexical item or phrase P.

Compositionality principles for compound phrases exemplified, say, by an imagined phrase structure ${}_A[B\ C]$ may be in effect as follows

$$\langle\!\langle {}_A[B\ C] \rangle\!\rangle \ = expr_A\ (\ \langle\!\langle B \rangle\!\rangle,\ \langle\!\langle C \rangle\!\rangle\ )$$

where $expr_A$ is an algebraic expression in ONTOLOG associated with the syntactical category A. Synonymous and near-synonymous noun phrases are to be mapped into the same or adjacent nodes (ontotypes) in the ontology.

Lexical and structural ambiguities give rise to algebraic sums in the resulting ontotype representing the phrase.

Contributions to the sum ontotype which are not ontotypologically well-formed with respect to the ontology reduce to null, thus disappearing in sums. In this way type restrictions are enforced on the semantics, e.g. ensuring use of admissible attributes, only.

The principles of an ontological semantics along this line is sketched in the following simple example.

The phrase

[a] travel to Sweden with dog

may be assigned the frame term

$\mathsf{travel} \times (dst : \mathsf{Sweden}) \times ((bmo + cum) : \mathsf{dog})$.

This term is located in the simplified ontological lattice as sketched:

$$\begin{array}{c}
\uparrow \\
\mathsf{event} \\
\uparrow \\
\mathsf{moving\ event} \\
\uparrow \\
\mathsf{travel} \\
\uparrow \\
\mathsf{travel} \times (dst : \mathsf{Sweden}) \\
\uparrow \\
\mathsf{travel} \times (dst : \mathsf{Sweden}) \times ((bmo + cum) : \mathsf{dog}) \\
\uparrow
\end{array}$$

The ontotypology forming part of the applied ontology may specify the admissible connections between ontotypes and role relations, thereby contributing to the disambiguation of the compound ontorelation (BMO + CUM). If the the instrumental role can be ruled out for dog in the ontotypology, only the companion-relationship CUM remains.

This ontological approach to semantics is elaborated further in [Jensen *et al.*] (present volume). It is similar in spirit to the generative lexicon approach in [Pustejovsky 95, Pustejovsky 98], but differs with respect to the chosen logical framework.

References

[Bach 86] E.Bach, The Algebra of Events, *Linguistics & Philosophy*, 9, 1986. pp. 5-16.

[Benthem & Meulen 97] J. van Benthem and A. ter Meulen, *Handbook of Logic and Language*, North-Holland, 1997.

[Braüner *et al.* 99] T. Braüner, J. Fischer Nilsson & A. Rasmussen: Conceptual Graphs as Algebras – with an Application to Analogical Reasoning, in Proceedings of the 7th Int. Conf. on Conceptual Structures, Blacksburg, Virginia, July 1999, W. Tepfenhart & W. Cyre (eds.) *Lecture Notes in Artificial Intelligence LNAI 1640*, Springer, 1999.

[Brink *et al.* 94] C. Brink, K. Britz, and R.A. Schmidt: Peirce Algebras, *Formal Aspects of Computing*, Vol. 6, 1994, pp. 339-358.

[Carpenter 92] B. Carpenter, *The Logic of Typed Feature Structures*, Cambridge University Press, 1992.

[Davey & Priestley 90] B.A Davey & H. A. Priestley, *Introduction to Lattices and Order*, Cambridge U.P., 1990.

[Eco 95] U. Eco: *The Search for the Perfect Language*, Blackwell, 1995.

[Jensen *et al.*] P. Anker Jensen, J. Fischer Nilsson & C. Vikner, Towards an Ontology-based Interpretation of NP's, present volume.

[Keenan 85] E.L. Keenan & L. M. Faltz, *Boolean Semantics for Natural Language*, Reidel, 1985.

[Larsen & Nilsson 97] H. Legind Larsen & J. Fischer Nilsson, Fuzzy querying in a concept object algebraic datamodel, in T. Andreasen *et al.* (eds.), *Flexible Query Answering Systems*, Kluwer, 1997.

[Link 98] G. Link: *Algebraic Semantics in Language and Philosophy*, CSLI Lecture Notes 74, Stanford, 1998.

[Madsen *et al.*] B. Nistrup Madsen, B. Sandford Pedersen and H. Erdman Thomsen, Defining Semantic Relations for OntoQuery, present volume.

[Nilsson 94] J. Fischer Nilsson: An Algebraic Logic for Concept Structures, *Information Modelling and Knowledge Bases V*, IOS Press, Amsterdam, 1994. pp. 75-84.

[Nilsson 99] J. Fischer Nilsson, A Conceptual Space Logic, in Proceedings of the European-Japanese Conference on Information Modeling and Knowledge Bases, Iwate, Japan, May, 1999, republished in E. Kawaguchi *et al.* (eds.) *Information Modelling and Knowledge Bases XI*, IOS Press/Ohmsha, Amsterdam, 2000. pp. 26-40.

[Pustejovsky 95] J. Pustejovsky, *The Generative Lexicon*, MIT Press, 1995, 1998.

[Pustejovsky 98] J. Pustejovsky, "Knowledge is Elsewhere": Natural Language Semantics meets the X-files, Brandeis University, 1998.

[Rounds 97] W. C. Rounds, Feature Logics, in [Benthem & Meulen 97]

[Simons 87] P. Simons, *Parts, a Study in Ontology*, Oxford, 1987.

[Somers 87] H. L. Somers, *Valency and Case in Computational Linguistics*, Edinburgh, 1987.

[Sowa 00] J.F. Sowa, *Knowledge Representation, Logical, Philosophical, and Computational Foundations*, Brooks/Cole Thomson Learning, 2000.

Appendix: Mathematical basis for Ontolog

This appendix briefly presents some mathematical background material for the ONTOLOG language.

Lattice Axioms

The intended logical functioning of the algebraic operators is established by introducing appropriate axioms in the form of algebraic equations.

The operations $+$ and $\times$ in lattice theory are known as *join* (supremum) and *meet* (infimum), cf. e.g. [Davey & Priestley 90], with the following axioms

Idempotency:

$$X + X = X$$
$$X \times X = X$$

Commutativity

$$X + Y = Y + X$$
$$X \times Y = Y \times X$$

Associativity

$$X + (Y + Z) = (X + Y) + Z$$
$$X \times (Y \times Z) = (X \times Y) \times Z$$

Absorption

$$X \times (X + Y) = X$$
$$X + (X \times Y) = X$$

A concept lattice is made distributive with the axiom

$$X \times (Y + Z) = (X \times Y) + (X \times Z)$$

and is bounded by the extremal elements univ (lattice top $\top$) and null (lattice bottom $\bot$) with

$$X + \mathsf{null} = X$$
$$X \times \mathsf{null} = \mathsf{null}$$
$$X + \mathsf{univ} = \mathsf{univ}$$
$$X \times \mathsf{univ} = X$$

Using these axioms it is possible to conduct rewritings (by hand or machine) on terms in ontologies and in this way perform logical inferences.

The axioms in the lattice algebra extended with binary relations are discussed in [Brink *et al.* 94, Nilsson 94, Nilsson 99].

Ontolog as logic

A general concept c is logically akin to a monadic predicate $c(\ )$, whereas an individual concept corresponds to a constant term.

Thus the inclusion of the general concept c' in c'', in predicate logic is expressed as

$$\forall x(c'(x) \rightarrow c''(x))$$

However, as a notational and conceptual complication in logic, for the case of c' being an individual concept, its inclusion in c'' is to be expressed as the atomic formula $c''(c')$ in predicate logic.

Similarly, disjointness of c' and c'' is expressed as $\neg\exists x(c'(x) \wedge c''(x))$ or equivalently

$$\forall x(c'(x) \wedge c''(x) \rightarrow \mathit{false})$$

The sample algebraic equations

$$\begin{aligned} c_{sup} &= c_1 + c_2 \\ c_{inf} &= c_1 \times c_2 \end{aligned}$$

in logic becomes

$$\begin{aligned} \forall x(c_{sup}(x) &\leftrightarrow c_1(x) \vee c_2(x)) \\ \forall x(c_{inf}(x) &\leftrightarrow c_1(x) \wedge c_2(x)) \end{aligned}$$

The Peirce product $(r : c)$ logically has the corresponding

$$\lambda x.\exists y(r(x,y) \wedge c(y))$$

Thus it is seen that the algebraic forms substantially simplify the corresponding logical specifications.

Ontolog as language of sets

According to lattice theory distributive lattices have sets as mathematical models.

Let us introduce $[\![c]\!]$ for the extension set of c.

Then the algebraic equations

$$\begin{aligned} c_{sup} &= c_1 + c_2 \\ c_{inf} &= c_1 \times c_2 \end{aligned}$$

corresponds to the set equations

$$[\![c_{sup}]\!] = [\![c_1]\!] \cup [\![c_2]\!]$$
$$[\![c_{inf}]\!] = [\![c_1]\!] \cap [\![c_2]\!]$$

In this extensional view, furthermore, the concept inclusion relationship $c' \leq c''$ expresses the subset relationship: $[\![c']\!] \subseteq [\![c'']\!]$.

The extension of $\perp$ is $\{\}$ and of $\top$ is the entire universe for the set extensions.

The correspondence between the ordering and the lattice operations also manifests itself as follows:

$$X \leq Y \quad \text{if and only if} \quad X = X \times Y \quad \text{if and only if} \quad Y = Y + X$$

which may be compared with the analogous connection for sets x, y, z:

$$x \subseteq y \quad \text{if and only if} \quad x = x \cap y \quad \text{if and only if} \quad y = y \cup x$$

The Peirce product $(r : c)$ set-theoretically reduces to

$$\{x | \exists y(\langle x, y\rangle \in [\![r]\!] \wedge y \in [\![c]\!])\}$$

Relationship to description logic and other systems

ONTOLOG constitutes an algebraic counterpart of a basic form of the so-called description logic. For an account of this relationship see further [Brink *et al.* 94]. A key advantage of the present algebraic form is that it explicitly supports lattices with accompanying diagrams having proved useful in ontological specification work. There are also strong affinities to feature logics [Carpenter 92, Rounds 97].

Lattice algebras are introduced as basis for a formal semantics for natural language with emphasis on determiners in [Keenan 85]. An algebraic approach is also adopted in [Bach 86] for describing events and in [Link 98] for establishing a formal semantics for plurals in particular.