# In Pursuit of Natural Logics for Ontology-Structured Knowledge Bases

Jørgen Fischer Nilsson

Department of Mathematics and Computer Science
Technical University of Denmark (DTU)
Email: jfni@dtu.dk

***Abstract*—We argue for adopting a form of natural logic for ontology-structured knowledge bases with complex sentences. This serves to ease reading of knowledge base for domain experts and to make reasoning and querying and path-finding more comprehensible. We explain natural logic as a development from traditional logic, pointing to essential differences to description logic. We conclude with a knowledge base set-up with an embedding into clausal logic, offering also a graph view of the sentences.**

***Index Terms*—Knowledge representation and reasoning; Ontologies; Natural language Interface.**

## I. Introduction

We address ontology-structured knowledge bases (KBs), that is KBs which encompass ontological classification structures as well as more general logical sentences. We outline a KB set-up which supports sentences in a regimented (controlled) fragment of natural language. This choice is motivated by our wish to make knowledge bases readable for domain experts. Moreover, this approach offers "generative ontologies" where linguistic terms generate new concept nodes in the ontology in addition to the given classes. In the devised meta logic set-up, the natural logic KB sentences are embedded in a clausal logic taking care of the inference and querying.

Our approach to ontological engineering is described further in recent papers [1][2][3][4]. We focus on knowledge bases within the life-sciences, which abound with complex textual descriptions and elaborate classification structures.

### A. State of the Art

Contemporary approaches to knowledge based systems aim at accommodating more complex information than admitted in traditional relational databases. The two competing prominent approaches are the rule-based representations in the form of logical clauses and various dialects of description logic. These logical representations are described and compared e.g. in Grosof et al. [5].

As an alternative to these logics for KBs we apply so-called natural logic for the considered KBs including formal ontologies. The applied natural logic possesses a logical semantics and is supported by reasoning rules applied directly to the natural logic form. Sentences in the natural logic resemble natural language, so that KBs can be read and understood by domain experts. Moreover, as a novelty, the set-up simultaneously provides a graph representation of the natural logic KB content as extension of the common ontological partial order classification diagrams. The supporting graph representation facilitates pathfinding in a KB. This functionality enables computation of shortest paths in the KB graph between user-stated concepts and entire phrases.

### B. The Structure of the Paper

The structure of this paper is as follows: In Section II, we take as departure traditional syllogistic logic. Then, in Section III we review *en passant* essentials of description logic as a tool for setting up formal ontologies. In Section IV, we turn to our main subject of natural logics, followed up, in Section V, by introduction of the natural logic fragment we propose for ontological knowledge bases. For the implementation set-up for the natural logic dialect we consider the logic of definite clauses in Section VI, which is used for embedding of the natural logic knowledge base in the devised KB systems design in Section VII. Finally, Section VIII concludes the paper.

## II. Traditional Syllogistic Logic and Ontologies

Let us begin recalling the Aristotelian natural logic syllogistic sentence forms [6] known from the square of opposition, see figure 1.

| | |
|---|---|
| every $C$ isa $D$ | no $C$ isa $D$ |
| some $C$ isa $D$ | some $C$ isa not $D$ |

Fig. 1 From the square of opposition.

Contemporary formal ontologies apply basically the class inclusion relation isa corresponding to the sentence form every $C$ isa $D$, which forms a partial order by way of reflexivity, transitivity and antisymmetry. Often, the partial order, rendered as a Hasse diagram (graph), simplifies to a classification tree with the universal top class at the root.

Although there are (unspecified) extension sets behind the classes, usually there is no requirement that the ontology has to form a distributive lattice, let alone a lattice by presence of supremum and infimum classes [7]. This is because the intensional comprehension of classes makes in particular many would-be union classes ontologically irrelevant [8].

As for the other three forms above in the square of opposition, in ontological engineering they are often expressed by introducing appropriate classes in the ontology, together with the default assumption that classes are disjoint if they have no common subclass and neither is a subclass of the other. Recall that traditional logic comes with existential import,

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-390-2 

meaning that there is no notion of empty classes. Individual concepts may formally be conceived of as singleton classes in the ontological set-up.

## III. Description Logic

Description logic (DL), the foremost contemporary knowledge representation logic, is a fragment of predicate logic. DL has become pivotal as logical basis for the semantic web research endeavors. The various description logic dialects share a variable-free algebraic form of expressions, with the general requirement that the logic is decidable with respect to desired functionalities, as well as tractable. In Grosof et al. [5], DL is compared to and aligned with the rule forms in definite clause logic. The tractability requirement implies that intended operations can be performed in polynomial time measured in terms of the size of the description logic specification. The various dialects of description logic differ by the admitted operators and the ensuing worst case complexity.

### A. Description Logic and Ontologies

The ontological class inclusion relationship "$C$ isa $D$" in description logic becomes

$$C \ \sqsubseteq \ D \tag{1}$$

In DL, there are no default rules such as the above mentioned existential import. Accordingly, disjointness of two classes is expressed, for instance by

$$C \sqcap D \ \equiv \ \bot \tag{2}$$

where $\bot$ is the predefined empty concept (class).

Classes $C$ and $D$ in DL generalize to various concept expression forms including set union, $\sqcup$, and intersection $\sqcap$. As such the ontological constitution in DL provides distributive lattices. Furthermore, even Boolean lattices are achieved by complement formation. In this way, DL offers class generativity in formal ontologies.

### B. Concept Modifiers

Description logic offers means of forming sub-classes (called concepts in DL) notably by means of a binary algebraic operator $\exists R.C$, where the first argument, $R$, is binary relation (a property in DL terminology), and the second one, $C$, is a concept expression. For instance, the concept of "*cells that produce insulin*", being a sub-concept of "*cell*", becomes

$$cell \sqcap \exists produce.insulin$$

From the point of view of ontological constitutions, the recursive syntactical form of the $\exists$ construct provides potentially unbounded generativity into ever more specialized concepts in the ontology.

Turning from concepts to entire assertions, the sample sentence "*cells that produce insulin reside in the pancreas*" in DL may become

$$cell \sqcap \exists produce.insulin \ \sqsubseteq \ \exists residein.pancreas$$

which seems hard to interpret for most domain experts, not the least because of the awkward 'subject-property' copula form, corresponding to "*cells and produce insulin are [something that] reside in the pancreas*".

## IV. Natural Logics

Natural logics are formal logics taking form of "regimented" fragments of natural language with accompanying inference rules for reasoning directly with the natural logic [9][10][11][12]. Quoting from the discussion of natural logic in [13]: "*The idea of the universality of logic is based on the conviction that [...] there are certain invariant features of human reasoning, carried out in any natural language whatsoever, that allow the formulation of universal logical laws, applicable to any language.*" In our setup, rather than translating the natural language forms into, say, DL, we conduct reasoning at the natural logic level, unlike Azevedo et al. [14].

### A. Class Relationships versus Property Ascriptions

Natural logics may be viewed as a development of traditional syllogistic logic continued via medieval logicians, e.g., John Buridan, see Klima [13][15], and via 19th century logicians, notably Peirce and De Morgan, see e.g. Sánchez Valencia [12]. A key point in this development is the abandoning of strict copula forms taking form of a subject and a predicate as in traditional syllogistic logic, in favour of logical sentences admitting a main verb expressing a binary relationship. In a more conceptual or ontological view, what is at stake here is acceptance of binary point relationships between classes rather than property ascription to classes. The latter "monadistic" view attributed to Leibniz, cf. [16], remains in DL.

As an example, in the property ascription view, informally the sentence *betacell produces insulin* is coined into the somewhat awkward *betacell isa (producer-of insulin)* possibly accompanied by the reciprocal *insulin isa (produced-by betacell)*.

## V. A Natural Logic For Knowledge Bases

Let us consider the natural logic in our [3][4] with sentences of the syntactic form

$$Q_1 \ C \ R \ Q_2 \ D \tag{3}$$

– where $Q_1$ and $Q_2$ are either of the determiners (quantifiers) every and some, and
– where $C$ and $D$ are nominal phrases, and
– where $R$ is a transitive verb.

In the simplest case, $C$ and $D$ are common nouns representing classes. These common nouns may next be adorned with modifiers in the form of linguistic relative clauses and adjectives. Modifiers are here assumed to act restrictively, unlike parenthetical relative clauses [13].

As it appears, the form (3) comprises four quantifier combinations, dubbed $\forall\forall, \forall\exists, \exists\forall, \exists\exists$ in [17]. From the point of view of ontological knowledge bases, the by far most common quantifier constellation among the 4 combinations is the $\forall\exists$ option

$$\text{every } C \ R \text{ some } D \tag{4}$$

Example: every betacell produces some insulin

– or, in short, using common default conventions as in natural language for this quantifier case:

betacell produce insulin

The existential quantification over substances such as insulin we ontologically understand as ranging over the conceivable amounts of the substance. These amounts constitute the imaginable individuals in a substance class.

One observes that the corresponding passive voice sentence [every] insulin isproducedby [some] betacell, with the reverse relation, is not logically equivalent [17]. However, the weaker some insulin isproducedby every betacell is entailed, adopting existential import throughout.

As an aside, one may notice that the considered form (4) fits perfectly with the partonomic forms in [18] with examples

[every] pancreas haspart [some] betacell

and

[every] betacell ispartof [some] pancreas.

### A. Inclusion in the Natural Logic and in Description logic

In the considered natural logic dialect, the class inclusion comes about with the above $\forall\exists$ form with the relation being equality. Thus

every betacell is-equal-to some cell

expresses the class inclusion

betacell isa cell.

This follows from a predicate logical explication of "every $C$ $equals$ some $D$" as $\forall x(C(x) \rightarrow \exists y(x = y \land D(y)))$, which is logically equivalent to $\forall x(C(x) \rightarrow D(x))$. We retain the distinguished short form isa relationship in our natural logic, since this relationship prevails in ontological knowledge bases.

The key natural logic sentence "every $C$ $R$ some $D$" in DL would become the somewhat awkward copula (subject predicate) form

$$C \sqsubseteq \exists R.D \tag{5}$$

cf. Section III.B.

### B. Compound Concept Terms

As mentioned, in the devised natural logic [2][3][4] class expressions may contain modifiers with restrictive relative clauses, as in the example

cell that produce insulin residein pancreas

contrast the DL formulation in Section III.B.

## VI. Definite Clause Logic: Rule Language

We now turn to definite clause rules as an additional component in our natural logic KB set-up. The building blocks of logical clauses are atomic formulas $p(t_1, ..., t_m)$, where $p$ is an $m$-argument predicate and the $t_i$ are logical terms. In the present context, these terms are confined to variables and constants representing individuals. Recall that the more general form of clauses applied in logic programming and artificial intelligence admits functional terms, consisting of a function symbol followed by term arguments.

As such, clausal logic specifies relationships between individuals, unlike the focus on relationships between concepts in syllogistic logic and description logic. This makes clausal logic *prima facie* unfit for ontology-structured knowledge bases dealing with relationship between classes.

A logical clause is a disjunction of atomic formulas or their denials, where all variables present (if any) are implicitly universally quantified. A definite clause is conveniently written and understood as an implication clause

$$p_0(t_{01}, ..., t_{0m_0}) \ \leftarrow \ \bigwedge_i^n \ p_i(t_{i1}, ..., t_{im_i}) \tag{6}$$

where the reverse implication arrow can be read as "if". The case of $n = 0$ yields an atomic formula, (called a fact if it is variable-free). Definite clauses where the terms are either variables or constants are known as DATALOG. A logical computation is initiated with an atomic formula as hypothesis to be confirmed or disconfirmed as logically entailed by the given clauses. The DATALOG logic enjoys properties of decidability (proved by propositionalization, i.e., reduction to propositional logic) and tractability, cf. [5].

Definite clauses only express assertive (positive) propositions. However, denials may be provided implicitly by the adoption of the closed world assumption, implying that the denial of a fact is taken to hold if the fact does not follow from the given clauses, a principle known as negation-by-failure (to prove). The DATALOG logic enjoys properties of decidability (proved by propositionalization, i.e., reduction to propositional logic) and tractability, cf. [5].

### A. Concepts Reified as Individuals

Definite clauses at the outset express relations between individuals as in

$$hormone(X) \leftarrow insulin(X)$$

However, by encoding of concepts as individuals definite clauses can emulate class-class relationships as in

$$isa(insulin, hormone)$$

supported by the clauses

$$isa^*(C, D) \leftarrow isa(C, D)$$
$$isa^*(C, D) \leftarrow isa(C, X) \land isa^*(X, D)$$
$$isa^*(C, C)$$

where $isa$ now represents the immediate (direct) subclass relationship, and the predicate name $isa^*$ its reflexive transitive closure computed in DATALOG. This encoding of classes suggests as a next crucial step embedding of the entire natural logic in clauses with supporting inference directly in the natural logic.

## VII. An Embedded Natural Logic

We now wish to embed the natural logic in DATALOG clauses acting as a metalogic for the natural logic. This calls for decomposition of natural logic sentences into atomic components which can be handled in DATALOG. We devise a decomposition that enables reconstruction of the natural logic compound sentences (modulo paraphrasing) [3][4].

As an instructive example, consider again the sentence

cell that produce insulin residein pancreas

This sentence in our system becomes decomposed into the following atomic ground (that is, variable-free) facts

$$isa(cell'that'produce'insulin, cell)$$
$$fact(definition,\; cell'that'produce'insulin, produce, insulin)$$
$$fact(observ,\; cell'that'produce'insulin, residein, pancreas)$$

where $cell'that'produce'insulin$ is formally an auxiliary individual constant in DATALOG, and simultaneously a fresh concept node label in the ontological graph view.

The two latter DATALOG facts, as it appears, comprise an epistemic mode tag. These modes affect the inference engine: The outlet definitions (including isa) of a concept node effectively act as an if-and-only-if definition, unlike the observational contributions. Further modes may be introduced, in order to distinguish normative, observational and hypothetical contributions.

This decomposition principle supports the graph view of ontologies with the subclass relationships forming the skeleton ontology, as it were.

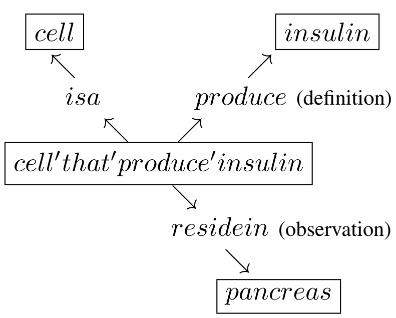


Fig. 2. Graph view of sample sentence.

Figure 2 shows the graph conception of the considered sentence, where the decomposition into the above three ground atomic facts appears as labeled, directed edges being outlet from the concept node $cell'that'produce'insulin$.

The epistemic distinctions ensure that the natural logic sentence is recoverable. They also ensure that relevant subsumption relationships can be computed and added to the KB [4]: Suppose that it is stated that insulin is a hormone in the KB ontology. Then the concept, say, $cell'that'produce'hormone$ is likely to occur also. In this case, the subsumption algorithm then is to compute and record the inclusion relationship $isa(cell'that'produce'insulin, cell'that'produce'hormone)$, concomitant with an additional arc in the graph. On the other hand, we refrain from pre-computing and storing those inclusion $isa$ relationships holding solely by virtue of transitivity, since the entire transitive closure relation would shortcut paths which might preferably be retained in pathway computations.

The compound natural logic sentences in the KB in general give rise to auxiliary nodes in the graph. And the graph contributions from the sentences form a single ontological KB graph with unique node representation of concepts across the sentence contributions. The original natural logic sentences can be reconstructed relying on the edge modes. Ideally, synonymic phrases such as pancreatic cell and cell that residein pancreas would be mapped into one concept node in the KB. One should also keep in mind that all the edges are here further assumed ∀∃-quantified.

### *A. Intensional Querying and Pathfinding*

The embedded knowledge base may now be queried deductively via the clause language, appealing to appropriate inference rules expressed as clauses.

Given class names, $c$, are introduced by

$$class(c)$$

The concepts (simple or complex) may be queried, say, with

$$\leftarrow isa^*(X, c)$$

giving for variable $X$ all concept terms below $c$,
– or more restrictively with

$$\leftarrow class(X) \wedge isa^*(X, c)$$

giving all subordinate class names.

The key inference rules in natural logic are the so-called monotonicity rules [9], which admits restriction of the grammatical subject concept to sub-concepts (recognized as inheritance), and, conversely, generalization of the grammatical object concept for the ∀∃ forms considered here [4]:

$$fact(M, Csub, R, Dsup) \leftarrow$$
$$isa^*(Csub, C) \wedge fact(M, C, R, D) \wedge isa^*(D, Dsup)$$

It follows logically, for instance, given cell that produce insulin residein pancreas and pancreas isa endocrinegland that cell that produce insulin residein endocrinegland.

In [2][3] we discuss pathway inference computations in natural logic KBs in the context of bio-models. This functionality aims at finding shortest paths in the KB graph between two stated concepts appealing to graph search algorithms. Pathway computations may formally be understood as application of a logical comprehension principle for composition of relations [1].

### *B. Class Disjointness in the Natural Logic*

As it stands, the present natural logic does not provide negation, unlike the classical negation available in DL. However, some form of negation is achievable by appeal to negation by non-provability in the rule logic, as known from logic programming and relational database querying.

In our set-up, two classes are considered disjoint unless one class is a sub-class of the other one, or that they have

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-390-2 

a common sub-class. We find this natural in ontological engineering, which often leans towards hierarchical classifications. Recall that classes here are assumed non-empty according to the principle of existential import. Then, overlap of two classes (concepts) can be ascertained with

$$overlap(C, D) \leftarrow isa^*(X, C) \wedge isa^*(X, D)$$

where the variable $X$ ranging over concept terms (including those stemming from the decomposition of sentences) may be considered existentially quantified to the right of the reverse implication.

Conversely the disjointness of two classes is verified with

$$disjoint(C, D) \leftarrow \text{NOT } overlap(C, D)$$

appealing to negation by non-provability, NOT, with the closed world assumption, conforming with use of negation in database query languages. From the point of view of ontology development use of the non-monotonic negation by non-provability implies that extension with new overlapping classes to the knowledge base may cancel out present class disjointness.

## VIII. Concluding Summary

We have advocated the adoption of forms of natural logic for ontology-structured knowledge bases in a set-up with embedding into definite clauses. This embedding of the natural logic sentences facilitates useful functionalities such as intensional reasoning and querying and pathway finding in large knowledge bases.

We conduct evaluation with a small scale prototype written in the logic programming language Prolog (supporting Datalog as a sublanguage) on life-science sample KBs in [2]. The prototype decomposes the natural logic sentences into the shown fact/graph KB representation. The devised decomposition of the natural logic sentences with inference rules in Datalog invites as a next development step large-scale implementation on relational data base platforms with the decomposed KB sentences represented as tuples.

## References


[1] T. Andreasen, H. Bulskov, J. Fischer Nilsson, P. Anker Jensen, and T. Lassen, “Conceptual pathway Querying of Natural Knowledge Bases from Text Bases”, 10th International Conference on Flexible Query Answering Systems, FQAS 2013, H. L. Larsen et al (ed.), LNAI 8132, Springer, 2013, pp. 1-12.

[2] T. Andreasen, H. Bulskov, J. Fischer Nilsson, and P. Anker Jensen, “Computing Pathways in Bio-Models Derived from Bio-Science Text Sources”, International Conference on Bioinformatics and Biomedical Engineering, F. Ortuño and I.Rojas (Eds.), IWBBIO 2014, April 7-9, Granada (Spain), 2014, pp. 217-226.

[3] T. Andreasen, H. Bulskov, J. Fischer Nilsson, and P. Anker Jensen, “Computing Conceptual Pathways in Bio-Medical Text Models”, Foundations of Intelligent Systems - 19th International Symposium, ISMIS 2014, Roskilde, Denmark, June 28-30, LNAI 8502, Springer, 2014, pp. 264-273.

[4] T. Andreasen and J. Fischer Nilsson, “A Case for Embedded Natural Logic for Ontological Knowledge Bases”, 6th International Conference on Knowledge Engineering and Ontology Development, KEOD2014, Rome, September 21st-24th, 2014, paper no. 71.

[5] B. N. G. Grosof, I. Horrocks, R. Volz, and S. Decker, “Description Logic Programs: Combining Logic Programs with Description Logic”, in Proceedings of the 12th international conference on World Wide Web (WWW ’03). ACM, New York, NY, USA, 2003, pp. 48-57.

[6] J. L. Ackrill (Ed.), Aristotle’s Categories and De Interpretatione, Oxford at the Clarendon Press, 1963.

[7] J. Fischer Nilsson, “Ontological Constitutions for Classes and Properties”, 14th Int. Conference on Conceptual Structures, ICCS 2006, Lecture Notes in Artificial Intelligence, LNAI 4068, Springer, 2006, pp. 37-53.

[8] D. M. Armstrong, A Theory of Universals: Volume 2: Universals and Scientific Realism, Cambridge Universty Press, 1978, 1990.

[9] J. van Benthem, Essays in Logical Semantics, Studies in Linguistics and Philosophy, Vol. 29, D. Reidel Publishing Company, 1986.

[10] J. van Benthem, “Natural Logic, Past And Future”, Workshop on Natural Logic, Proof Theory, and Computational Semantics 2011, CSLI Stanford, URL: http://web.stanford.edu/˜icard/logic&language/2011.NatLog.pdf [accessed: 2015-01-20].

[11] R. Muskens, “Towards Logics that Model Natural Reasoning, Program Description Research Program in Natural Logic”, URL: http://lyrawww.uvt.nl/˜rmuskens/natural/ NLprogram.pdf [accessed: 2015-01-20].

[12] V. Sánchez Valencia, “The Algebra of Logic”, in D. M. Gabbay and J. Woods (Eds.), Handbook of the History of Logic, Vol. 3 The Rise of Modern Logic: From Leibniz to Frege, Elsevier, 2004.

[13] G. Klima, “Natural Logic, Medieval Logic and Formal Semantics”, URL: http://filozofiaiszemle.net/wp-content/uploads/2012/06/Gyula-Klima-Natural-Logic-Medieval-Logic-and-Formal-Semantic.pdf [accessed: 2015-01-20].

[14] R. R. de Azevedo, F. Freitas, R. Rocha, J. A. Menezes, and L. F. A. Pereira, “Generating Description Logic $\mathcal{ALC}$ from Text in Natural Language”, in Proceedings of Foundations of Intelligent Systems - 21th International Symposium, ISMIS 2014, Roskilde, Denmark, June 25-27, LNAI 8502, Springer, 2014, pp. 305-314.

[15] G. Klima, John Buridan, Oxford University Press, 2009.

[16] J. Fischer Nilsson, “On Reducing Relationships to Property Ascriptions”, Information Modelling and Knowledge Bases XX, Volume 190 Frontiers in Artificial Intelligence and Applications, (Eds.) Y. Kiyoki *et al.*, IOS Press, Hardcover ISBN: 978-1-58603-957-8, 2009, pp. 245-252.

[17] J. Fischer Nilsson “Diagrammatic Reasoning with Classes and Relationships”, A. Moktefi and S.-J. Shin (Eds.), Visual Reasoning with Diagrams, Studies in Universal Logic, Birkhäuser, Springer, pp. 83-100, 2013.

[18] B. Smith and C. Rosse, “The Role of Foundational Relations in the Alignment of Biomedical Ontologies”, MEDINFO 2004, M Fieschi et al. (Eds.), Amsterdam IOS press, pp.444-448, 2004.