A System for Conceptual Pathway Finding and Deductive Querying

Troels Andreasen, Henrik Bulskov, Jørgen Fischer Nilsson, and Per Anker Jensen

Abstract We describe principles and design of a system for knowledge bases applying a natural logic. Natural logics are forms of logic which appear as stylized fragments of natural language sentences. Accordingly, such knowledge base sentences can be read and understood directly by a domain expert. The system applies a graph form computed from the input natural logic sentences. The graph form generalizes the usual partial-order ontological sub-class structures by accommodation of affirmative sentences comprising recursive phrase structures. In this paper we focus on the logical inference rules for extending the concept graph form enabling deductive querying as well as computation of pathways between the concepts mentioned in the sentences.

Keywords Deductive querying of natural-logic knowledge bases · Path finding in knowledge bases · Logical knowledge bases in bio-informatics and medicine

1 Introduction and Background

In a series of papers [1, 2, 4, 5, 6] we have recently developed and described principles and systems design for natural-logic knowledge bases. This work originates in our

T. Andreasen(✉) · H. Bulskov
Computer Science, Roskilde University, Roskilde, Denmark
e-mail: {troels,bulskov}@ruc.dk

J.F. Nilsson
Mathematics and Computer Science, Technical University of Denmark, Kongens Lyngby, Denmark
e-mail: jfni@dtu.dk

P.A. Jensen
International Business Communication, Copenhagen Business School, Frederiksberg, Denmark
e-mail: paj.ibc@cbs.dk

T. Andreasen et al. (eds.), *Flexible Query Answering Systems 2015*,
Advances in Intelligent Systems and Computing 400,
DOI: 10.1007/978-3-319-26154-6_35

idea of providing so-called generative ontologies [7]. In our generative ontologies, the concepts are not merely given classes but entire phrases in which the class noun is extended with restrictions for forming subclasses. These restrictive phrases, as in the natural language phrases they reflect and formalize, are endowed with a recursive structure, thereby becoming "generative", in analogy to the well-known notion of generative grammars.

In the above-mentioned more recent papers we go a step further by adopting a simplified form of so-called natural logic [8, 9] as our formal language for stating propositions. Accordingly, a knowledge base (KB) consists of a finite set of affirmative sentences in natural logic. These sentences comprise traditional ontological sub-class relationships as special cases, so there is no separate formal ontology. As discussed in our [1] the natural logic formulations come close to natural language so that the KB can be read by domain experts, for instance, in the bio-sciences. It goes without saying that the formal natural logic dialect cannot accommodate the full meaning content of a text sentence in natural language. As discussed in our [6], it is our contention that semantically the considered natural logic can cover substantial parts of typical textual specifications within the bio-sciences.

In the present paper, we focus on computing conceptual pathways between concept terms stated as a query. In order to achieve this functionality, we have devised a graph form of the knowledge base in which the possibly complex knowledge base sentences are broken down into more elementary ones without essential loss of meaning. As part of this endeavour, we address the deductive querying of natural-logic knowledge bases.

The paper is structured as follows: In section 2 we describe the applied natural logic with the accompanying internal graph form in section 3. In section 4 we describe the inference rules applying to the graph form and brought to bear on pathway querying in section 5.

2 Natural Logic for Knowledge Bases

The knowledge base sentences considered express relationships between classes in an ontology. The applied form of natural logic is meant to be readable for domain experts without background in logic and computer science. At the same time, the considered natural logic dialect constitutes a well-defined logic as discussed in [3, 4, 5].

2.1 Simple Sentences in Natural Logic

The simplest sentence form

Cnoun isa *Cnoun*

expresses class inclusion. *Cnoun*-expressions are common nouns naming introduced classes. The knowledge base ontology is shaped by such sentences, where the class inclusion relationship forms a partial ordering of the classes. As an example, we may

have betacell isa cell. Notice that the system is incapable of splitting agglutinated compounds like "betacell" in order to identify a head noun, in casu "cell".

More generally, the logic admits knowledge base sentences with transitive verbs

Cnoun Verb Cnoun

as in the example betacell produce insulin. Thus, in addition to the strictly ontological class inclusion structure, the knowledge base comprises more general state-of-affairs descriptions.

2.2 Complex Sentences in Natural Logic

Crucially, we further admit compound, recursively structured class terms

Cterm Verb Cterm

as in the sample

cell that produce insulin located:in pancreatic gland,

where the phrase cell that produce insulin denotes a sub-class (complex concept) of the given class cell formed by a restrictive relative clause. Similarly, the adjectival modifier "pancreatic" introduces a subclass of the class "gland". Generally speaking, the various types of modifiers always act restrictively in the set up.

Restrictive relative clauses may recursively comprise restrictive relative clauses as in the phrase gland that haspart (cell that produce hormone). The parentheses here are for clarification, only. Thus, in principle, an open-ended and unrestricted collection of classes is made available, although in a knowledge base with accompanying queries, obviously, only a finite set would be made explicit. This notion of generative ontologies was launched in a seminal form in [7]. The various suggested language forms are further described in our [1, 4, 5]. Sample knowledge bases are found in our [6].

2.3 The Logical Understanding of Sentences

From a logical point of view, all the knowledge base sentences *Cterm Verb Cterm* are implicitly quantified, namely as

every *Cterm Verb* some *Cterm*

giving for instance every betacell produce some insulin as explicitation of the above betacell produce insulin. As is evident, there are actually four possible quantifier constellations in the above sentence form. However, in this context we only consider the above quantifier form, since it covers substantial parts of the knowledge base information in the considered applications. This is confirmed by the default assumptions applied in natural language concerning this adopted quantifier form.

The natural logics offer an alternative to description logics. Specifically, the natural logics recognize the key role of the main verb in natural language affirmative sentences. This is in contrast to description logics, where sentences come about as extended copula forms, hampering the human comprehension of knowledge bases. For instance, the sample, straightforward sentence betacell produce insulin in description

logic becomes the rather incomprehensible betacell $\sqsubseteq \exists$produce.insulin as discussed in [4].

In [1, 4, 5], we discuss further the relationships to syllogistic logic, predicate logic, and description logic. There we also discuss our approach to denials by way of a default assumption amounting to considering classes disjoint unless one is a subclass of the other or they have a common subclass. More generally, we lean towards the closed world assumption, unlike the open world assumption of description logic.

3 The Concept Graph Form

The logical view of sentences supported by inference rules described below affords deductive query facilities. In our system complex sentences are decomposed into simple sentences. The simple sentences are thought of as arcs in a labeled directed graph called the concept graph. Crucially, the decomposition of complex sentences calls for generation of new concept nodes in the graph corresponding to the concept terms as well as any sub-terms in the knowledge base sentences.

The graph view complements the logical view of knowledge base sentences by affording computational path finding between - possibly complex - concepts. We elaborate on the functionalities offered by the graph view in the final sections of this paper.

As an example consider again the sentence cell that produce insulin located:in pancreatic gland. In our system, this given sentence becomes decomposed into the simple sentences:

cell-that-produce-insulin isa cell
cell-that-produce-insulin produce insulin
cell-that-produce-insulin located:in pancreatic-gland

where cell-that-produce-insulin is a system-generated concept term which names a node as illustrated in figure 1. Since adjectival modifications are always taken for

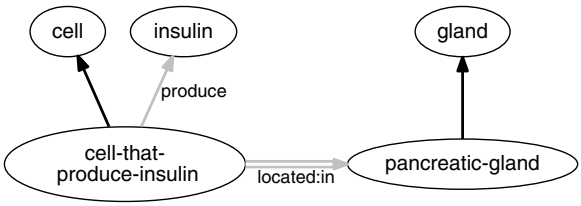

Fig. 1 Graph representation of the sentence "cells that produce insulin are located:in the pancreatic gland"

being restrictive here, the system adds pancreatic-gland isa gland.

In order to ensure that the meaning of a sentence in the knowledge base is properly retained in the graph, we distinguish three different arcs as illustrated in figure 1. The arcs contributing to the definition of a complex concept are drawn as single arrows.

isa-arcs are drawn as black arrows, whereas restrictive contributions to the definition are drawn as grey arrows. The arc stemming from the verb in the main sentence, which creates the proposition, is drawn as a double arrow.

The representation of concepts is assumed to be unique and thus shared across the contributing sentences. Accordingly, the KB sentences give rise to one, usually coherent, graph.

4 Inference Rules

The considered sentences are subject to logical inference rules, that is, inference rules provided for purely logical reasons with reference to the underlying predicate logical explication. These rules admit deductive querying of the knowledge base.

In addition, there may be ad hoc rules supporting introduced relationships cf. the example in section 4.3.

4.1 Logical Inference Rules

First and foremost, the isa relation is made reflexive and transitive, that is, a partial order:

$$\frac{}{C \text{ isa } C}$$

$$\frac{C \text{ isa } X \qquad X \text{ isa } D}{C \text{ isa } D}$$

As a simple example, given the two KB sentences: pancreatic-gland isa endocrine-gland and endocrine-gland produce hormone, we conclude that pancreatic-gland produce hormone using the inheritance rule:

$$\frac{C\ R\ D \qquad C' \text{ isa } C}{C'\ R\ D}$$

Moreover, given that betacell produce insulin and insulin isa hormone we conclude that betacell produce hormone using the rule of property generalization:

$$\frac{C\ R\ D \qquad D \text{ isa } D'}{C\ R\ D'}$$

These two rules are known as monotonicity rules in natural logic. As it appears they express common sense reasoning without appeal to complicated logical inference systems such as resolution and natural deduction.

The transitivity, inheritance and generalisation rules are illustrated in the figures 2 to 4. The inferences drawn by these rules are not materialized in advance in the concept graph. A stated query like betacell produce hormone? is confirmed by appeal

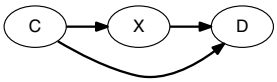

Fig. 2 Transitivity

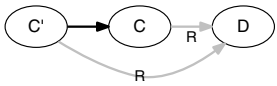

Fig. 3 Inheritance

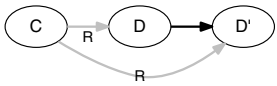

Fig. 4 Generalization

to the last of the above inference rules. Thus, derived sentences are not computed in advance.

4.2 The Subsumption Rule

The use of decomposed sentences in the KB concept graph calls for a special logical inference rule, termed the subsumption rule. This rule is to ensure that all logically relevant isa class inclusion arcs, less those following by transitivity of isa, become present in the graph.

As an example consider the two concept terms

cell-that-produce-hormone and cell-that-produce-insulin

giving rise to

cell-that-produce-hormone isa cell
cell-that-produce-hormone produce hormone

respectively

cell-that-produce-insulin isa cell
cell-that-produce-insulin produce insulin

and assume that the proposition insulin isa hormone is included ind the KB. In this case, as illustrated in figures 5 and 6, the subsumption rule calculates

cell-that-produce-insulin isa cell-that-produce-hormone

The subsumption pre-computation thus calculates missing class inclusion arcs, and thereby serves to facilitate and crucially speed up subsequent deductive reasoning computations and pathway computations in the concept graph. In some cases, the calculation would have to take regress to inclusion arcs throughout the concept graph. Therefore, we devise the following algorithm for systematically calculating the missing inclusion arcs. The algorithm relies on the principle that all inclusion arcs drawn on in a specific case have already been calculated.

The first step is to rank the concept nodes in the graph according to a depth criterion: Concept nodes which have no non-isa outlet arcs in their definitions are assigned order 0. Concept nodes whose non-isa outlet arcs lead to concept nodes of order 0 are assigned the order 1. Concept nodes whose non-isa outlet arcs lead to concept nodes of order n (and in addition possibly less) are assigned the order $n+1$. It should be noted that there is no risk of cycles in the definition graph, assuming that there are no cycles in the isa inclusion sub-graph.

The ranking of concept nodes is to ensure that when a pair of concept nodes is checked for subsumption, all the concept nodes pointed to from this pair have already been processed. Accordingly, the algorithm begins with all ranks up to 1 pairs of concept nodes in the entire graph and processes these.

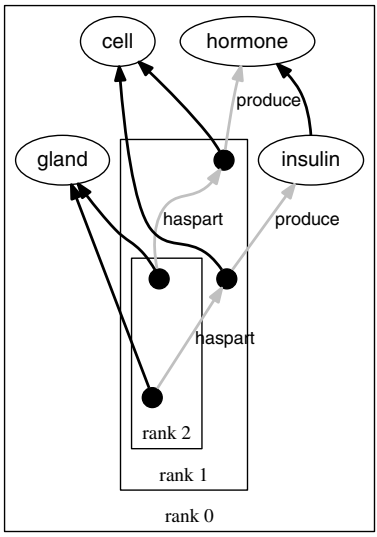

Fig. 5 Before addition

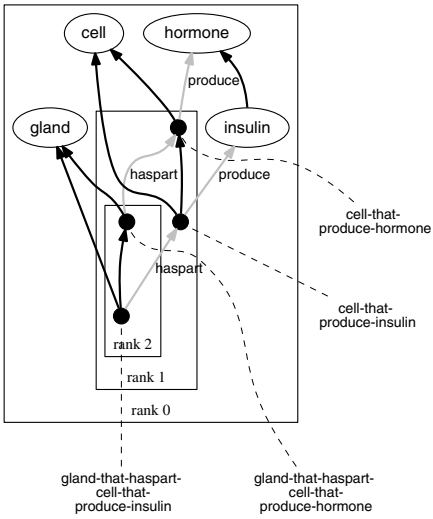

Fig. 6 After addition of subsumption arcs

Consider all pairs of nodes C and C' of rank 1, where

C has arcs $C\ R_i\ D_i$ for $i = 1..m$

and

C' has arcs $C'\ R_i\ D'_i$ for $i = 1..n$

where the sets of arcs $R_i\ D_i$ and $R_i\ D'_i$ may include inherited arcs according to inheritance inference, cf. figure 3. Now, assume that for all $R_i\ D'_i$ there is $R_i\ D_i$ so that D_i isa D'_i, either explicitly or by transitivity. In that case, add the arc C isa C'.

The algorithm then proceeds to up to rank 2 pairs of concept nodes (less the pairs having already been processed) and processes these, knowing that the concept nodes pointed to have already been processed. The algorithm continues in this way up to the highest rank being used in the concept graph. An example showing addition of missing arcs at rank 1 as well as rank 2 is illustrated in figures 5 (before) and 6 (after). One should observe that the highest rank is not given statically simply by the syntactic depth nesting of phrases in the original propositions.

4.3 Domain Dependent Inference Rules

As an example of a domain inference rule the has-part relation may be made transitive (cf. [10]) by way of the rule:

$$\frac{C \text{ haspart } X \qquad X \text{ haspart } D}{C \text{ haspart } D}$$

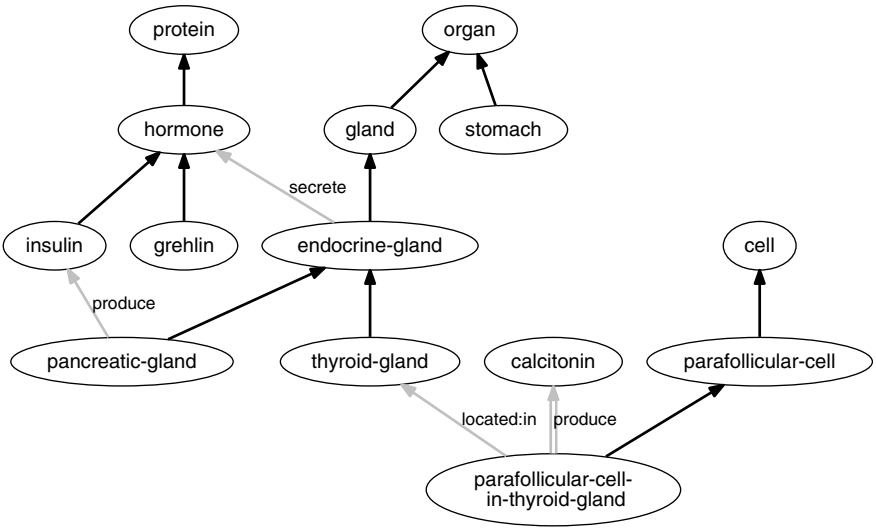

Fig. 7 A fragment of an ontology centered around endocrine gland

Similarly for the complementary part-for relation. Again, these rules are to be activated in the KB system rather than be used for pre-computation of derived relationships.

5 Concept Path Finding

The concept graph is a logical view of the sentences in the corpus from which it has been generated. Sentences are decomposed into simple sentences that correspond to edges in the graph defining concepts and expressing propositions. Thus, a path in the concept graph, a concept path, combines a series of simple sentences and may therefore be rendered in natural language into an explanation of the connection between the end nodes of the path. Concept path finding can thus be applied as a means of knowledge base querying. Given two or more concepts, we can search for natural-language renderings of connections relating these. Given a single concept, we can search for related key concepts. We thus consider queries to reveal connectivity in the graph. Below we mainly consider two-concept queries.

As mentioned above, we assume that the concept graph G is closed wrt subsumption, such that all edges that are inferable by the subsumption rule, are included in G.

Candidate answers to a two-concept query $Q = (C, C')$ are based on paths connecting the two query concepts C and C' or, more specifically, paths connecting C to C'. From any such path we can derive a natural-language rendering corresponding to the connection it provides. For instance, an answer to the query $Q =$ (pancreatic-gland, protein) evaluated on a knowledge base corresponding to the graph in figure 7 involves the path:

(pancreatic-gland produce insulin), (insulin isa hormone), (hormone isa protein) or more succinctly:

(pancreatic-gland produce insulin isa hormone isa protein)

From this we can derive the natural-language rendering:

pancreatic-gland produce insulin, which is a hormone, which is a protein.

All edges in the concept graph are directed. However, not only directed paths may contribute to answers to two-concept queries. Given a two-concept query $Q = (C, C')$ we consider in principle any undirected path from C to C'. Thus the direction of edges does not influence the paths we are considering, but only the interpretation and thereby the natural language rendering we can apply on these.

5.1 Rendering a Path Into Natural Language

A natural language rendering of a path can be provided as follows. The rendering of an inclusion edge in the beginning of the path X isa Y is "X, *which is a* Y", while an inner edge that continues from a previous node and leads to Z on the path simply adds ", *which is a* Z" to the rendering. Thus the rendering of the path X isa Y isa Z will be "X, *which is a* Y, which is a Z".

When in the beginning of the path, an inclusion edge traversed in the inverse direction, for instance, a path from Z to Y through an edge Y isa Z, can be read as "*some* Z *are* Y", while an inverse inclusion inner edge that continues the path from a previous node and leads to X on the path adds ", *whereof some are* X" to the rendering. Thus, the rendering of the path from Z through Y to X provided by the two edges Y isa Z and X isa Y will be "*some* Z *are* Y, *whereof some are* X".

Semantic relations (i.e. relations other than isa) are named by the main verb in the phrase from which they are extracted, and these may therefore be read "as is". Thus the rendering of the forward direction of an edge $X\ R\ Y$ beginning a path is simply "$X\ R\ Y$" , while an inner edge that continues a path can be read "$R\ Y$". As with the inclusion relation, semantic relations may be traversed in the inverse direction. However, for semantic relations we assume explicitly specified inverse relations. For a relation R the inverse relation is given by $\bar{R} = inv(R)$, where inv is a symmetric mapping given by a domain expert.

When in the beginning of the path, an edge corresponding to the relation R traversed in the inverse direction, for instance, a path from Z to Y through an edge $Y\ R\ Z$, can be read as "*some* Z *are* $inv(R)$ Y", while an inverse semantic inner edge that continues the path from a previous node and leads to X on the path adds ", *whereof some are* $inv(R)$ X" to the rendering. Thus, for instance, the rendering of the path from Z through Y to X provided by the two edges $Y\ R\ Z$ and $X\ R\ Y$ will be "*some* Z *are inv(R)* Y, *whereof some are inv(R)* X.

As an example, an answer to the query $Q =$ (protein , pancreatic-gland) evaluated on figure 7 based on the path indicated above in inverse direction would lead to the rendering:

some protein are hormone, whereof some are insulin, whereof some are produced by pancreatic gland.

assuming that inv(produce) = produced:by, while an answer to Q = (protein , gland) would lead to:

some protein are hormone, whereof some are secreted by endocrine gland, which is a gland.

assuming that inv(secrete) = secreted:by.

5.2 Reduction

Among the potentially most interesting paths that may be applied to provide answers to a query $Q = (C, C')$, are the shortest paths connecting the two query concepts C and C'. However, due to the fact that a significant number of conceptual edges derivable by the inference rules are not explicitly included in the graph G, we cannot be sure that a shortest path between C and C' in G provides the briefest connection between the two concepts. A path connecting two concepts C and C' may be reduced, replacing edges according to inference, such that premise edges are removed and inferred edges are inserted. Due to the transitivity inference rule, a path or a subpath may be reduced by edge replacement

$$(C \text{ isa } X \text{ isa } D) \qquad \text{replaced by} \qquad (C \text{ isa } D)$$

Similarly we can derive possible replacements from the two monotonicity inference rules. Due to inheritance monotonicity, a path or a subpath may be reduced by replacing edges:

$$(C' \text{ isa } C\ R\ D) \qquad \text{replaced by} \qquad (C'\ R\ D)$$

and due to generalization monotonicity, a path or a subpath may be reduced by:

$$(C\ R\ D \text{ isa } D') \qquad \text{replaced by} \qquad (C\ R\ D')$$

Thus, by applying generalisation twice or transitivity followed by generalisation, we can reduce

(pancreatic-gland produce insulin isa hormone isa protein) to
(pancreatic-gland produce protein)

while by applying inheritance followed by generalization, we can reduce

(pancreatic-gland isa endocrine-gland secrete hormone isa protein) to
(pancreatic-gland secrete protein)

The shortest path in figure 7 connecting *calcitonin* and *protein* (assuming *inv* (produce) = produced:by) is the following:

(calcitonin produced:by parafollicular-cell-in-thyroid-gland
located:in thyroid-gland isa endocrine-gland secrete hormone isa protein)

This path may be reduced to

(calcitonin produced:by parafollicular-cell-in-thyroid-gland
located:in endocrine-gland secrete protein)

Reduction leads to shorter paths and thereby to more succinct natural-language renderings of connectivity. This is obviously at the expense of details which in some cases may provide useful supplementary information. Thus a possibility in a user interface to expand reduced paths to their original form would be a useful feature making a more dynamic interface.

An alternative less coarse-grained reduction principle could also be applied: always retain nodes that have outgoing semantic relation edges (relations other than isa). This would correspond to ignoring inheritance while reducing paths.

5.3 Query Evaluation Principle

Evaluating two-concept queries to a concept graph is first of all a matter of finding paths in the graph. The principle indicated above and described in more detail below divides into shortest path computation, reduction and natural-language rendering. The path computation applies a Breadth First Search (BFS) starting from the first query concept.

Given the directed concept graph G and assuming that G is closed wrt subsumption. Let $\bar{G}$ be an undirected version of G and let the query $Q = (C, C')$ be a two-concept query.

1. Derive the set P of all shortest paths from C to C' in $\bar{G}$. Start from C, apply BFS continuously adding all new paths from C to the set B until C' is found, return $P = \{p | p \in B,\ p\ connects\ C\ and\ C'\}$
2. For each path $p \in P$ derive p' by repeatedly reducing subpaths until no further reduction can be performed, set $P = P \cup \{p'\}$
3. Let $\sigma = min(\{l | p \in P,\ l = length\ of\ p\})$
4. Let $\bar{S} = \{p | p \in P, \sigma = length\ of\ p\}$
5. Let S be the set of paths in G corresponding to the paths $\bar{S}$ in $\bar{G}$
6. For all $p \in S$ provide the rendering of p as contribution to the answer to Q

It should be noted that we cannot ensure that all shortest paths will be found by this algorithm. Continuing step 1 until all paths are found may result in additional paths that can be reduced to a shortest path in step 2. There will also be cases where this will lead to a shorter length of the shortest paths found.

6 Summary and Future Work

We have outlined a system for pathfinding in logical knowledge bases. The key component in the system is a concept graph being pre-computed from a given knowledge

base which consists of sentences in natural logic. In computing the concept graph we strive – if only heuristically, so far – to strike a balance between "materialized" information in the form of stored arcs versus virtual information deducible by means of the stated inference rules. As the guiding principle we require that all "isa" class inclusion relationships except for those following by transitivity are explicitly recorded. Therefore, the described subsumption algorithm is to be invoked in a pre-computation phase. On the other hand, we refrain from pre-computing the entire transitive closure of the class inclusion as well as what follows from applying the monotonicity rules. Currently we are performing small-scale experiments with a prototype.

References

1. Andreasen, T., Bulskov, H., Nilsson, J.F., Jensen, P.A.: A system for computing conceptual pathways in bio-medical text models. In: Andreasen, T., Christiansen, H., Cubero, J.-C., Raś, Z.W. (eds.) ISMIS 2014. LNCS, vol. 8502, pp. 264–273. Springer, Heidelberg (2014)
2. Andreasen, T., Bulskov, H., Nilsson, J.F., Anker Jensen, P., Lassen, T.: Conceptual pathway querying of natural logic knowledge bases from text bases. In: Larsen, H.L., Martin-Bautista, M.J., Vila, M.A., Andreasen, T., Christiansen, H. (eds.) FQAS 2013. LNCS, vol. 8132, pp. 1–12. Springer, Heidelberg (2013)
3. Nilsson, J.F.: Diagrammatic reasoning with classes and relationships. Moktefi, A., Shin, S.-J. (eds.) Visual Reasoning with Diagrams. Studies in Universal Logic. Birkhäuser, Springer (2013)
4. Nilsson, J.F.: In pursuit of natural logics for ontology-structured knowledge bases. In: The Seventh International Conference on Advanced Cognitive Technologies and Applications, COGNITIVE 2015, Nice, France, March 22–27, 2015. IARIA. ISSN: 2308–4197, ISBN 978-1-61208-390-2
5. Andreasen, T., Nilsson, J.F.: A case for embedded natural logic for ontological knowledge bases. In: Proceedings of the 6th International Conference on Knowledge Engineering and Ontology Development (2014)
6. Andreasen, T., Bulskov, H., Nilsson, J.F., Jensen, P.A.: Computing pathways in bio-models derived from bio-science text sources. In: IWBBIO 2014, pp. 217–226 (2014)
7. Andreasen, T., Nilsson, J.F.: Grammatical Specification of Domain Ontologies in journal. Data & Knowledge Engineering **48**(2), 221–230 (2004)
8. van Benthem, J.: Essays in Logical Semantics. Studies in Linguistics and Philosophy, vol. 29. D. Reidel Publishing Company (1986)
9. van Benthem, J.: Natural logic, past and future. In: Workshop on Natural Logic, Proof Theory, and Computational Semantics 2011. CSLI Stanford (2011). http://www.stanford.edu/~icard/logic&language/index.html
10. Smith, B., Rosse, C.: The role of foundational relations in the alignment of biomedical ontologies. In: Fieschi, M. (ed.) MEDINFO 2004 (2004)