Deductive Querying of Natural Logic Bases

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

Deductive Querying of Natural Logic Bases

Troels Andreasen[1], Henrik Bulskov[1], Per Anker Jensen[2], and Jørgen Fischer Nilsson[3]

[1] Computer Science, Roskilde University, Denmark {troels, bulskov}@ruc.dk
[2] Management, Society and Communication, Copenhagen Business School, Denmark paj.msc@cbs.dk
[3] Mathematics and Computer Science, Technical University of Denmark, Denmark jfni@dtu.dk

Abstract. We introduce a dedicated form of natural logic intended for representation of sentences in a knowledge base. Natural logic is a version of formal logic whose sentences cover a stylized fragment of natural language. Thus, the sentences in the knowledge base can be read and understood directly by a domain expert, unlike, say, predicate logic and description logic. The paper describes the inference rules enabling deductive querying of the knowledge base. The natural logic sentences and the inference rules are represented in Datalog providing a convenient graph form. As such, the natural logic knowledge base may be viewed as an enriched formal ontology structure. We describe various query facilities including pathway finding accommodated by this setup.

Keywords: Natural Logic · Knowledge Bases · Deductive Querying · Formal Ontology

1 Introduction

This paper briefly introduces a form of natural logic [Sánchez Valencia, 1991], [van Benthem, 1986], [Klíma, 2010] intended for ontology-structured knowledge bases. After having introduced a natural logic dubbed NaturaLog, the second half of the paper focuses on deductive querying of natural logic knowledge bases. Basically the natural logic states quantified relationships between classes. As such, NaturaLog generalizes the inclusion relationships in formal ontologies by admitting arbitrary relations in addition to class inclusion. In addition, the present version of NaturaLog offers complex class terms reflecting the phrase structures of natural language according to the principles of generative ontologies, cf. [Andreasen and Nilsson, 2004]. Furthermore, NaturaLog bears some affinity to description logic as discussed in [Andreasen et al., 2018]. However, as one noteworthy difference, NaturaLog abandons forced use of copula sentences in favor of sentences with free choice of transitive verbs from the considered knowledge base domain.

We explain how NaturaLog sentences can be encoded as propositions in Datalog, that is definite clauses without function symbols. Moreover, the inference rules used for deductive querying are also expressed in Datalog extended, where relevant, with negation as non-provability as known from logic programming. Representing the sentences and the inference rules in Datalog ensures decidability and tractability of the deductive querying.

In sections 2 and 3 we explain the elementary forms of NaturaLog sentences and the encoding of NaturaLog as propositions. We proceed in section 4 by explaining the relevant inference rules for the purpose of deductive querying as Datalog clauses. Thereafter, in section 5 we extend NaturaLog to comprise recursively structured noun phrases and verbs endowed with adverbial forms. This extension calls for decomposition of the compound phrases in the Datalog Encoding. Section 6 explains the concept of materialization and section 7 demonstrates various different query facilities. Finally, we conclude in section 8.

The present paper focuses on query functionalities for NaturaLog knowledge bases and follows up on our former publications on the application of natural logic in [Andreasen et al., 2017a], [Andreasen et al., 2015], [Andreasen et al., 2017b], [Nilsson, 2015]. The details of the syntax and the formal semantics of the NaturaLog logic are elaborated in the coming paper [Andreasen et al., 2018] that also specifies NaturaLog in terms of predicate logic and introduces the concomitant graph conception of NaturaLog knowledge bases.

2 Elementary NaturaLog Sentence Forms

NaturaLog elementary sentences primarily take the form

[every] $Cnoun$ $Verb$ [some] $Cnoun$

where $Cnoun$ is a common noun and $Verb$ is a transitive verb (i.e. a verb taking a linguistic object), or $Verb$ may be the copula *is* written isa as common in formal ontologies. In the natural logic, we ignore inflections, using singular forms of common nouns.

With the indicated defaults for the determiners (quantifiers), as an example there is

betacell produce insulin

for

every betacell produce some insulin

As another example one may state the inclusion

insulin isa hormone

for

every insulin is-equal-to some hormone

understood modelling-wise as claiming that every portion of insulin is-equal-to some portion of hormone. A knowledge base consists of a finite number of affirmative NaturaLog sentences. A NaturaLog knowledge base may be conceived as a class-inclusion ontology extended with more general sentence forms containing transitive verbs stating relations between classes. The determiner *no* yielding denials is not admitted in knowledge base sentences. It appears only in deduced sentences by way of the closed world assumption.

In [Andreasen et al., 2018] we discuss alternative quantifiers to the given defaults

(every|some|no) $Cnoun$ $Verb$ (some|every) $Cnoun$

3 Encoding of NaturaLog as Propositions

We now describe the encoding of NaturaLog sentences in Datalog since this is crucial for deductive querying. Recall that Datalog restricts clauses to being definite logical clauses without compound terms, so that predicate arguments are either implicitly universally quantified variables or constants. Logically, NaturaLog sentences represent relationships between two relata classes or concepts with the inclusion relation as a common case in formal ontologies, cf. [Smith and Rosse, 2004], [Moss, 2010].

The above elementary NaturaLog sentences are encoded as what we refer to as propositions in Datalog factual atomic sentences of the form

$\mathbf{p}(Det, Noun, Verb, Noun)$

with a predicate **p** at this metalogical level, and where Det is either every or some. At the Datalog metalogical level $Noun$ and $Verb$ appear as constants as in the sample

p(every,betacell,produce,insulin)

In the propositions at the metalogical level nouns are conceived of as concepts C (one-argument predicates) and verbs as relations R. For the default case of the determiner being every, we introduce a variant form of the predicate **p** through the pair of defining clauses

p(C,R,D) $\leftrightarrow$ **p**(every,C,R,D)

In the setup of the knowledge base nouns and verbs are declared by additional metalevel predicates as in the sample

concept(betacell), **concept**(insulin), **relation**(produce), **relation**(isa)

or indirectly by

concept(C) $\leftarrow$ **p**(Q,C,R,D)
concept(D) $\leftarrow$ **p**(Q,C,R,D)
relation(R) $\leftarrow$ **p**(Q,C,R,D)

For a full predicate logical construal of NaturaLog we refer to [Andreasen et al., 2018].

We appeal to the principle of existential import, implying that there is no explicit presence of an empty concept, cf. [Andreasen et al., 2017a,Nilsson, 2015]. In section 4, we explain how this doctrine may handle the case of disjoint concepts, that is, concepts having an empty overlapping concept. This principle means that all concepts appearing in a knowledge base proposition are assumed to be non-empty through presence of a hypothetical anonymous entity. By contrast, concepts that do not appear anywhere in the knowledge base are assumed to be empty in the deductive querying process.

4 Inference Rules

The universally quantified variables appearing in the following Datalog clauses effectively range over encoded concepts C, D etc. and relations R in NaturaLog propositions.

Reflexivity

$\mathbf{p}(\mathsf{C, isa, C})$

Monotonicity

$\mathbf{p}(\mathsf{C', R, D}) \leftarrow \mathbf{p}(\mathsf{C', isa, C}) \wedge \mathbf{p}(\mathsf{C, R, D})$

$\mathbf{p}(\mathsf{C, R, D'}) \leftarrow \mathbf{p}(\mathsf{C, R, D}) \wedge \mathbf{p}(\mathsf{D, isa, D'})$

Transitivity

$\mathbf{p}(\mathsf{C, R, D}) \leftarrow \mathbf{trans}(\mathsf{R}) \wedge \mathbf{p}(\mathsf{C, R, CD}) \wedge \mathbf{p}(\mathsf{CD, R, D})$

Inversion (passive voice formation)

$\mathbf{p}(\mathsf{some, D, Rinv, C}) \leftarrow \mathbf{inv}(\mathsf{R, Rinv}) \wedge \mathbf{p}(\mathsf{some, C, R, D})$

Weakening of quantifiers

$\mathbf{p}(\mathsf{some, C, R, D}) \leftarrow \mathbf{p}(\mathsf{C, R, D})$

Denials

$\mathbf{p}(\mathsf{no, C, R, D}) \leftarrow \nvdash \mathbf{p}(\mathsf{some, C, R, D})$

Disjointness

$\mathbf{p}(\mathsf{no, C, isa, D}) \leftarrow \nvdash \mathbf{common}(\mathsf{C, D})$

$\mathbf{common}(\mathsf{C, D}) \leftarrow \mathbf{p}(\mathsf{CD, isa, C}) \wedge \mathbf{p}(\mathsf{CD, isa, D})$

The auxiliary predicate **inv** lists pairs of inverse relations such as **inv**(promote, promoted_by).

The above inference rules serve logical purposes being justified eventually by the underlying logical construal of NaturaLog. In addition, one may introduce ad hoc rules such as transitivity for causality and parthood relations.

5 Restrictive Modifiers

We now extend the elementary NaturaLog sentences by incorporation of recursive phrase structures acting as restrictive modifiers. When adjoined to nouns modifiers provide subconcepts (specialization) of the concept denoted by the noun. Similarly, when adjoined to verbs as adverbials they yield subrelations (specialized relations).

This paper focusses on adnominal modifiers, assuming in the logical treatment that they linguistically take the form of restrictive relative clauses as in the sample noun phrase

cell that produce hormone

A more comprehensive treatment of the linguistic variants would include also adnominal PPs and restrictive adjectives.

Restrictive modifiers give rise to formation of auxiliary concepts that are subconcepts of the head noun concept. These auxiliary concepts may be thought to come about by synonymity as if instituted by the pair

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

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

However, these prospective sentences are shown for explanatory reasons only, as they are going to be dealt with in a dedicated manner: At the metalogical level of propositions modifiers are handled by means of definitions complementing the propositions **p** and being of the factual clausal form

$\mathbf{d}(Caux, C, R, C')$

with the additional predicate **d** (for definition) and where $Caux$ is a new concept coming about by modification of the concept C with the modifier that $R\ C'$, as in that produce hormone. As a special case, R may be isa. The new concept $Caux$ is generated as a constant in the metalogical representation as in the sample

d(cell-that-produce-hormone, cell, produce, hormone)

Such a definition is made to act as two propositions through

p(CRC, isa, C) $\leftarrow$ **d**(CRC,C,R,C')
p(CRC, R, C') $\leftarrow$ **d**(CRC,C,R,C')

The definition **d**(CRC',C,R,C') can be visualized as in figure 1, where by convention unlabelled arcs represent isa.

With the stated inference rules and adding a rule covering subsumption,

p(X, isa, CRC') $\leftarrow$ **d**(CRC', C, R, C') $\wedge$
p(X, isa, C) $\wedge$ **p**(X, R, C')

and assuming **p**(X,isa,C) and **p**(X,R,C') for any concept X, one gets the inferred proposition as indicated by the dashed arrow in figure 2(a). The subsumption rule is to be activated in a compilation phase in order to ensure that all inclusion relationships are made explicit prior to querying.

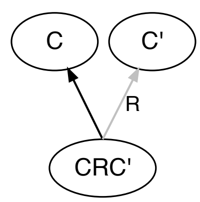


Fig. 1. Definition of the concept CRC'

For definitions, the defining edges share origins as shown for CRC' in figure 1.

As an example consider the definition **d**(cell-that-produce-insulin, cell, produce, insulin). From this the two propositions **p**(cell-that-produce-insulin, isa, cell) and **p**(cell-that-produce-insulin, produce, insulin) follows. Now given **p**(betacell,produce,insulin) and **p**(betacell, isa, cell), it follows by subsumption that **p**(betacell, isa, cell-that-produce-insulin), as indicated in figure 2(b).

The syntactic class of noun phrases with restrictive modifiers further comprises conjunctions as shown schematically in

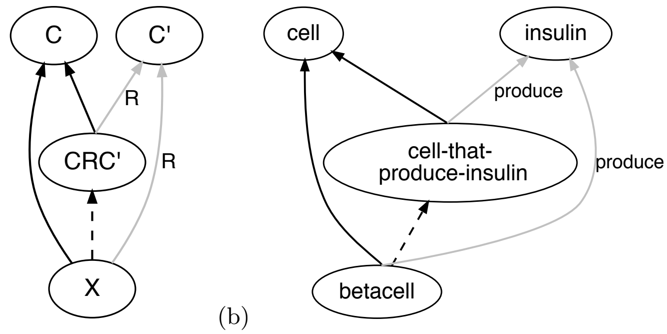
C that $R1$ $C1$ and C that $R2$ $C2$

giving rise to

d(C-that-$R1$-$C1$, C, $R1$ $C1$)
d(C-that-$R2$-$C2$, C, $R2$ $C2$)

and in turn

d(C-that-$R1$-$C1$-and-C-that-$R2$-$C2$, C-that-$R1$-$C1$, isa, C-that-$R2$-$C2$)



(a) (b)

Fig. 2. Propositions inferred by subsumption (dashed edges)

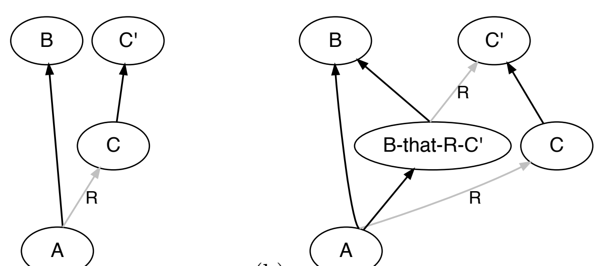
Example: cell that produce hormone and cell that in thyroid_gland
This should not to be confused with the recursively nested form in
production of hormone in thyroid_gland

6 Materialization of Concepts in the Knowledge Base

We advance the following subsumption materialization principle:

> Besides those concepts that are mentioned in the deconstructed knowledge base propositions, all conceivable concepts that subsume those concepts are to be materialized in the knowledge base. Moreover, these materialized concepts are furnished with their pertinent isa relations (less, generally, isa relations following by transitivity) and coalesced into the knowledge base.

In other words, the ontology logically inherent in the knowledge base by virtue of isa-relationships is to be completed upwards by adding the necessary, finite number of additional concepts like B-that-R-C' in figure 3 or lack-of-hormone in figure 4.



(a) (b)

Fig. 3. Materialization of a new concept B-that-R-C' from a defined concept A

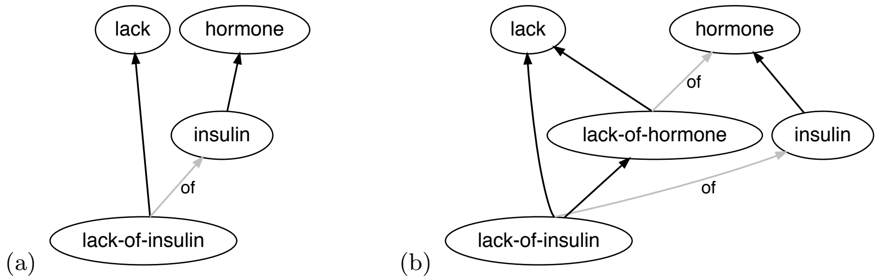


Fig. 4. Materialization of a new concept lack-of-hormone from a defined concept lack-of-insulin

As a corollary to this principle, all non-empty concepts, that is, all concepts possibly contributing to query answers, are made explicit in the knowledge base in advance by a compilation of the entire knowledge base. Thus, the subsumption materialization principle ensures that concepts that may appear as query answers are explicitly present in the knowledge base as an integral part of the ontological structure. The original individual propositions remain unaffected by this compilation. Inference rules for dynamically creating these new concepts and fitting them into the ontology are described in our [Andreasen et al., 2018].

7 Querying

In this section, we discuss some of the query functionalities for NaturaLog. Given that the NaturaLog knowledge base consists of propositions encoded as Datalog atomic clauses supported by the stated clausal inference rules in Datalog, querying can now be explained and carried out as deduction initiated by an appropriate query clause as known from logic programming.

7.1 Concept Querying

The primary query case is a NaturaLog sentence containing a variable as in

X isa c_{query}

being encoded as **p**(X,isa, c_{query}), where X is a Datalog variable ranging over all concept terms, and where the constituent symbols of c_{query} are assumed to be present in the knowledge base. For instance, considering the knowledge base fragment in figure 2, the term c_{query} could be the concept cell that produce insulin with the expected deduced answer being betacell, while, considering figure 6, an answer to a query hormone would be insulin.

Assuming that the term c_{query} is actually present in the knowledge base, it is to provide as answers all concept terms immediately subsumed by c_{query}, that is, residing just below c_{query}. Effectively, then, the answer is trivial if the term c_{query} itself is the only one available due to the reflexivity of isa.

7.2 Relaxed Concept Querying

Now, let us turn to the case where c_{query} is absent from the knowledge base. Then, logically the answer is empty in the closed world assumption setting. However, in order to achieve a more flexible query functionality, we devise a relaxation principle transcending deduction in DATALOG amounting to ascending step by step in the ontological structure from where the concept c_{query} would have been be placed. As an example, take the query concept

cell that reside_in brain and that produce insulin

In a first ascending step, given that brain isa organ and insulin isa hormone, this would relax to

cell that reside_in organ and that produce insulin

as well as to

cell that reside_in brain and that produce hormone

both potentially subsuming concepts in the knowledge base, thereby providing useful answers by subsumption.

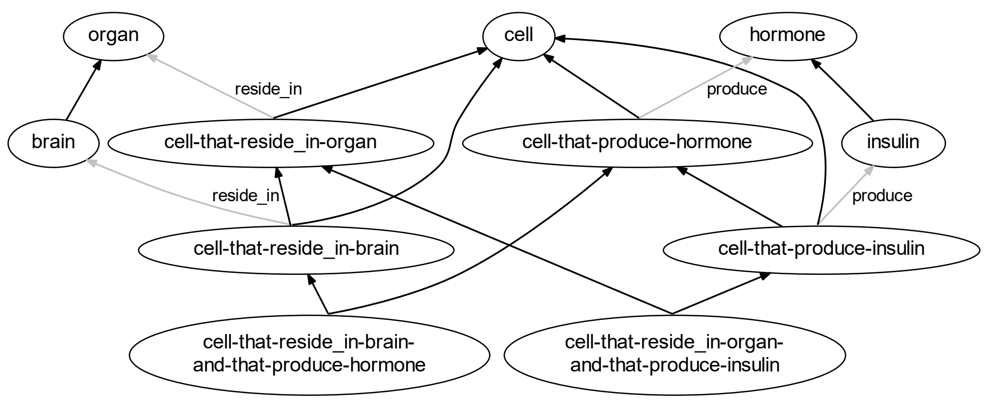


Fig. 5. Relaxed query where the query term is absent from the knowledge base

In outline, the relaxation algorithm works as follows: Without essential loss of generality, assume that c_{query} is of the form c-that-r_1-c_1-and-that-r_2-c_2 with the two restrictive modifiers that-r_1-c_1 and that-r_2-c_2. Then consider the three concept terms formed by generalization

c^{sup}-that-r_1-c_1-and-that-r_2-c_2

c-that-r_1-c_1^{sup}-and-that-r_2-c_2

c-that-r_1-c_1-and-that-r_2-c_2^{sup}

where c^{sup}, c_1^{sup} and c_2^{sup} are the corresponding concept terms one step up ontologically (assuming here for simplicity that they are unique). If either of these is present in the knowledge base, then those ones are engaged for deducing the query answers as shown in section 7.1. Otherwise the relaxation step is iterated one step up again.

7.3 Pathway Querying

The entire knowledge base graph forms a road map between all the applied concepts. The introduction of a universal concept at the top of the ontology ensures that all concepts are connected. This concept map can be queried by means of rules searching pathways in the graph between two stated concepts as sketched here:

$$\textbf{path}(C, D) \leftarrow \textbf{p}(Q, C, R, CD) \wedge \textbf{path}(CD, D)$$
$$\textbf{path}(C, D) \leftarrow \textbf{p}(Q, C, R, D)$$

In our setup, the knowledge base may be conceived as a bidirectional graph: by applying the inversion and the weakening rules and leaving the quantifiers Q unspecified, the predicate **path** may exploit the inverse relation paths. The interesting pathways are obviously the shortest ones, employing appropriate distance weights to the various relations.

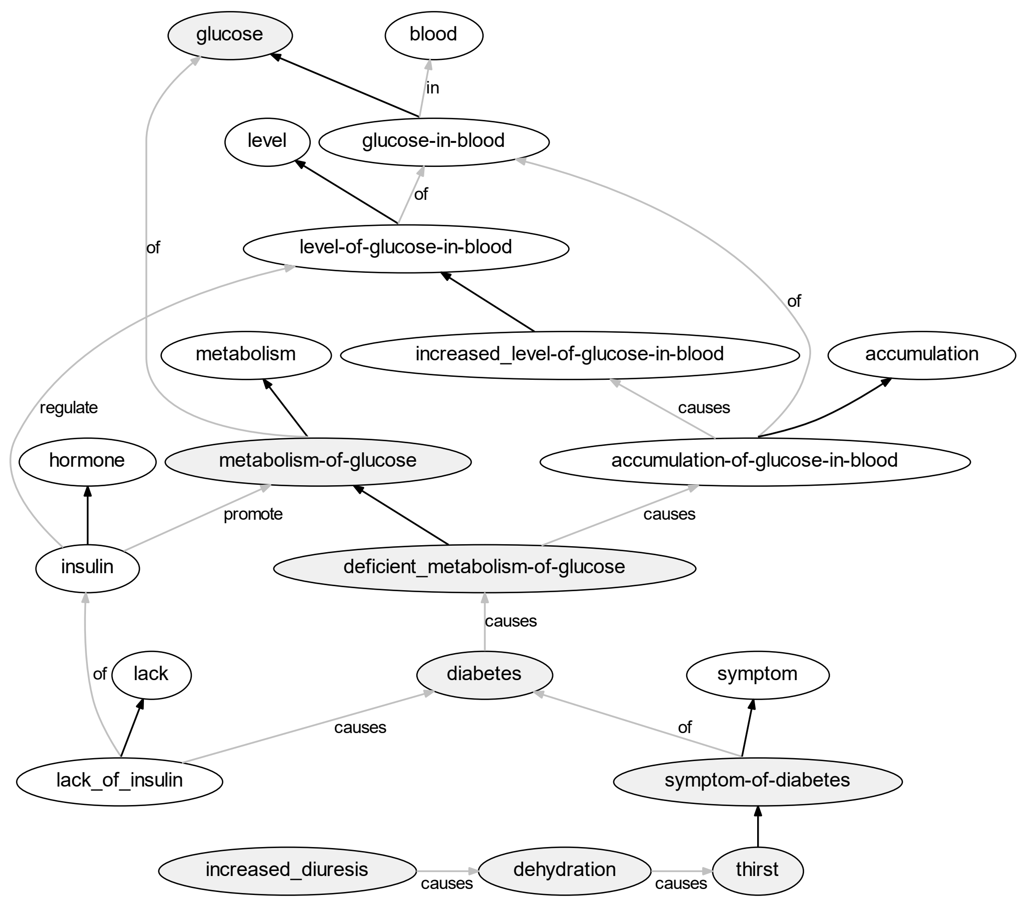


Fig. 6. Knowledge base fragment exemplifying a pathway (greyed nodes)

Below we consider a few examples of pathway queries with reference to the knowledge base fragment shown in figure 6, loosely based on [Kehler, 1988], [Weston et al., 1997].

The pathway query (diabetes, accumulation) requesting a path connecting the two concepts. This would correspond to the edges traversed by evaluation of **path**(diabetes, accumulation), that is, the following propositions

p(diabetes, causes, deficient_metabolism-of-glucose),
p(deficient_metabolism-of-glucose, causes, accumulation-of-glucose-in-blood),
p(accumulation-of-glucose-in-blood, isa, accumulation)

thus, a path connecting diabetes and accumulation is

diabetes – causes – deficient_metabolism-of-glucose – causes –
accumulation-of-glucose-in-blood – isa – accumulation

An example of a pathway query exploiting inverse relation paths is (deficient_metabolism-of-glucose, insulin). One possible answer path would involve the following propositions

p(deficient_metabolism-of-glucose, isa, metabolism-of-glucose),
p(some, metabolism-of-glucose, promoted_by, insulin)

Notice that the latter proposition traverses an edge in the opposite direction by applying the weakening rule leading to **p**(some, insulin, promote, metabolism-of-glucose) and then the inversion inference rule assuming the fact inv(promote, promoted_by). Thus, the connecting path in this case is

deficient_metabolism-of-glucose – isa –
metabolism-of-glucose – promoted_by – insulin

As a final example consider the query (increased_diuresis, glucose). One path connecting the two query concepts would be based on the following propositions

p(increased_diuresis, causes, dehydration),
p(dehydration, causes, thirst),
p(thirst, isa, symptom-of-diabetes),
p(symptom-of-diabetes, of, diabetes),
p(diabetes, causes, deficient_metabolism-of-glucose),
p(deficient_metabolism-of-glucose, isa, metabolism-of-glucose),
p(metabolism-of-glucose, of, glucose),

and thus traverse

increased_diuresis – causes – dehydration – causes – thirst – isa – symptom-of-diabetes – of – diabetes – causes – deficient_metabolism-of-glucose – isa – metabolism-of-glucose – of – glucose,

The corresponding path is indicated in figure 6 by the greyed nodes.

Notice that derived paths can be reduced by applying again the inference rules. For instance increased_diuresis – causes – dehydration – causes – thirst can be reduced to increased_diuresis – causes – thirst assuming **trans**(causes).

8 Conclusion and perspectives

We have described a deductive knowledge base setup for a natural logic based on the DATALOG decidable logic used as metalogic.

In addition to the query forms described in this paper, we envisage query facilities such as spreading activation traversing near concepts in the graph, commonality querying deducing common properties of two concepts and analogy queries, where, given a, b and c, an analogy query is to deduce an X such that $X\ R\ c$ whenever we have $a\ R\ b$.

References

[Andreasen et al., 2015] Andreasen, T., Bulskov, H., Jensen, P. A., and Nilsson, J. F. (2015). A system for conceptual pathway finding and deductive querying. In *Flexible Query Answering Systems 2015*, pages 461–472. Springer.

[Andreasen et al., 2017a] Andreasen, T., Bulskov, H., Jensen, P. A., and Nilsson, J. F. (2017a). *Partiality, Underspecification, and Natural Language Processing*, chapter A Natural Logic for Natural-Language Knowledge Bases. Cambridge Scholars.

[Andreasen et al., 2017b] Andreasen, T., Bulskov, H., Jensen, P. A., and Nilsson, J. F. (2017b). *Pathway Computation in Models Derived from Bio-Science Text Sources*, pages 424–434. Springer International Publishing, Cham.

[Andreasen et al., 2018] Andreasen, T., Bulskov, H., Jensen, P. A., and Nilsson, J. F. (2018). Natural logic knowledge bases and their graph form. 34 pages, submitted to journal (under review).

[Andreasen and Nilsson, 2004] Andreasen, T. and Nilsson, J. F. (2004). Grammatical specification of domain ontologies. *Data Knowl. Eng.*, 48(2):221–230.

[Kehler, 1988] Kehler, A. (1988). *Anatomi og Fysologi I og II.* Arnold Busk.

[Klíma, 2010] Klíma, G. (2010). Natural logic, medieval logic and formal semantics. *Logic, Language, Mathematics*, 2.

[Moss, 2010] Moss, L. S. (2010). Syllogistic logics with verbs. *J. Log. Comput.*, 20(4):947–967.

[Nilsson, 2015] Nilsson, J. F. (2015). In pursuit of natural logics for ontology-structured knowledge bases. In *The Seventh International Conference on Advanced Cognitive Technologies and Applications.*

[Sánchez Valencia, 1991] Sánchez Valencia, V. M. (1991). *Studies on Natural Logic and Categorial Grammar.* Categorial grammar. Universiteit van Amsterdam, Amsterdam, Holland.

[Smith and Rosse, 2004] Smith, B. and Rosse, C. (2004). The role of foundational relations in the alignment of biomedical ontologies. *Medinfo*, 11(Pt 1):444–448.

[van Benthem, 1986] van Benthem, J. (1986). *Essays in Logical Semantics, Volume 29 of Studies in Linguistics and Philosophy.* D. Reidel, Dordrecht, Holland.

[Weston et al., 1997] Weston, T., Horton, C., and Weston, T. (1997). *Atlas of Atonomy.* Marshall Cavendish.