On the Design of a Natural Logic System for Knowledge Bases

Troels Andreasen[1(✉)], Henrik Bulskov[1], and Jørgen Fischer Nilsson[2]

[1] Computer Science, Roskilde University, Roskilde, Denmark
{troels,bulskov}@ruc.dk

[2] Mathematics and Computer Science, Technical University of Denmark,
Lyngby, Denmark
jfni@dtu.dk

Abstract. Natural logics are logics that take form of stylized natural language sentences within a selected fragment of natural language. Natural Logics are at the same time formal logics with a well-defined syntax and semantics. Therefore, natural logics may be advanced as knowledge base logics enhancing explainability of query answers. This paper is concerned with a natural logic, NaturaLog, having been proposed as a deductive knowledge base language. The paper briefly reviews and brings together in compact form the main points in our previously but separately published design proposals, systems functionalities and implementation principles.

Keywords: Natural logic · Knowledge base systems · Deductive querying · Life science applications

1 Introduction

In a historical perspective there are two development lines in logic, that is logic-of-language and mathematical logic. The logic-of-language tradition dates back to Aristotle and went through developments during the medieval times until the end of the 19th century. This development line was halted and largely abandoned by the advent of quantified predicate logic due mainly to G. Frege and B. Russell. Predicate logic and related logics, then, have become foundational concepts as well as important tools in in computer science, in particular in computational logic, e.g. with logic programming.

However, the logic-of-language tradition recently has attracted renewed interest in connection with attempts to ease communication with computers. This paper discusses natural logics [9,12] that are rooted in the logic-of-language tradition and describes undertakings aimed at adopting and adapting natural logics for logical knowledge base systems. The following sections briefly surveys and discusses the various aspects of design principles and implementation methods described in the here chronologically listed range of papers [1–5,10,11] followed up by [6,7] forthcoming in 2020.

D. Helic et al. (Eds.): ISMIS 2020, LNAI 12117, pp. 413–421, 2020.
https://doi.org/10.1007/978-3-030-59491-6_39

Our design concern throughout is to obtain a useful trade-off between expressivity and computational tractability, having in mind also the requirements for potentially useful application domains in the design. Although the proposed natural logic is meant as a general purpose specification language for real-world domains, we have had in mind in particular applications within the life sciences as it appears in the mentioned publications.

2 Designing a Natural Logic

Our natural logic proposal, termed NaturaLog, takes as point of departure syllogistic logic from the Aristotelian tradition, cf. [8,11]. This means that the basic so-called categorical sentence forms are every C is D and some C is D, where C and D are class- or concept terms. In the simplest cases C and D are common nouns representing classes aka concepts. Accordingly, these forms known as copula forms express, respectively, a subclass relationship and a class-class overlap relationship. In the following, for the copula "is" we follow the conventions in computer science and write "isa". With the form every C isa D (or in convenient short form simply C isa D), one can specify hierarchically- as well as non-hierarchically structured formal ontologies as partial orders, with the isa relationship being transitive.

In addition to these affirmative sentence forms the old syllogistic logic also comprises negative forms as mentioned in [11], also known from the square-of-opposition. However, at present we refrain from admitting negative statements in a knowledge base itself, resorting instead to negation-as-non-provability as known from databases and logic programming.

2.1 Beyond Copula Forms: General Relationships

In addition to the copula isa in NaturaLog one can introduce transitive verbs (i.e. verbs taking a linguistic object) as one pleases. To this end in [2,3,11] we introduced the more general sentence forms with a verb R

$$(\textsf{every} \mid \textsf{some})\ C\ R\ (\textsf{every} \mid \textsf{some})\ D$$

giving four determiner constellations. The verb R represents a binary relationship between the two concepts represented by the subject and object. As convenient default form for the most common sentences in knowledge bases we propose

$$C\ R\ D \quad \text{for the full form} \quad \textsf{every}\ C\ R\ \textsf{some}\ D$$

Example: persons like pets is shorthand for every person likes some pet and some persons drink beer is shorthand for some persons drinks some beer. Actually, in NaturaLog we ignore linguistic inflection rules as seen in the following examples. For the some form we have

$$\textsf{some}\ C\ R\ D \quad \text{for} \quad \textsf{some}\ C\ R\ \textsf{some}\ D$$

In predicate logic every C R some D is construed as

$$\forall x(Cx \rightarrow \exists y(Rxy \wedge Dy))$$

and some C R some D is

$$\exists x(Cx \wedge \exists y(Rxy \wedge Dy)), \text{ which is equivalent to } \exists x \exists y(Cx \wedge Dy \wedge Rxy)$$

However, we stress that NaturaLog sentences are not translated into predicate logic in our systems proposal as explained in Sect. 3 and 4.

2.2 Compound Terms

In addition to the concepts given by common nouns in NaturaLog one can form expressions for creation of new concepts by attachment of restrictive modifiers to common nouns as in the sample sentence

betacell isa cell that produce insulin

The restrictive modifiers may take form of relative clauses as in the compound concept term cell that produce insulin or prepositional phrases such as in cell in gland. Both forms semantically consist of a relationship given as a verb or as a preposition followed by a concept. Provision is made for nesting of such constructs reflecting the usual recursive syntax for modifiers in natural language phrases.

There are other forms of restrictive nominal modifiers in natural language, in particular adjectives (including participles such as "increased") and noun-noun-compounds such as "heart disease". The incorporation of these modifiers into natural logic is more problematic and is postponed since they, unlike the above ones, do not directly provide a modifying relationship. As a temporary solution noun-noun compounds may be rewritten using a relative clause, so that for instance "bacteria infection" would become infection that is-caused-by bacteria.

Verbs may also be modified restrictively using an adverbial prepositional phrase as in the verb form produce in gland as a restriction of the verb produce. Incorporation of this useful feature, which falls outside the simple predicate logical explanation in Sect. 2.1, is thoroughly discussed in [6].

The syntax for the current version of NaturaLog is specified in the form of a BNF grammar in [6]. In order to ensure that there be no structural ambiguities, parentheses are enforced in one production rule for stipulating the intended recursive phrase structure.

In [3] we discuss various language extensions intended for promoting the usability of NaturaLog by approaching some common forms in natural language. Examples are appositions and conjunctions. We distinguish semantically conservative extensions and non-conservative ones. The former ones do not extend the semantic coverage. The order of the natural logic sentences in a knowledge base is logically irrelevant; the sentences are syntactically independent of each other.

2.3 Active-Passive Voice and Existential Import

Now, consider the active voice sentence betacell produce insulin or in full form every betacell produce some insulin. The corresponding passive voice sentence is some insulin is-produced-by some betacell, where is-produced-by represents the inverse relation of produce. Although this latter sentence follows intuitively, the sentence does not follow logically, because in predicate logic the denotation of a monadic predicate may well be empty. This problem is overcome by appealing to the existential import principle known from the Aristotelian logic tradition, cf. [6,11]. This principle declares that all mentioned classes be non-empty without being specific about any member entity. As a special consequence, the presence of a copula sentence of the form [every] C isa D implies availability also of the weakened converse sentence form some D isa C.

2.4 Remarks on Natural Logic and Description Logic

Todays most common logics for ontologies and knowledge bases are presumably the various description logic dialects. Both NaturaLog and description logics are examples of variable free logics covering small but useful fragments of predicate logic. A key difference between these two logics is that description logics offer sentences in copula form only (at the so-called T-box level of concepts), which seems awkward from the point of conventional use in natural language. This is in disagreement with common formulations in natural language. Another difference is that description logics have to resort to awkward reformulations for sentences beginning with the determiner "some", cf. the discussion of active-passive forms in the previous section. A further comparison of the two logics is given in [6].

The endorsing in the natural logic of non-copula sentences (with verbs fetched from the target application) agrees well with an entity-relationship model view of a knowledge base: A NaturaLog knowledge base typically takes the form of an ontology formed by the stated copula sentences extended with non-copula sentences connecting concepts across the ontology with relations expressed by transitive verbs. This view further invites the introduction of a distinction between definitional and observational (i.e. empirical) statements mentioned in the next section.

3 The Metalogic Framework for NaturaLog

So far, NaturaLog may be conceived simply as a "sugared" fragment of predicate logic for describing the application domain of discourse. As a next step appropriate proof rules admitting computational derivation of logical consequences as NaturaLog sentences are to be introduced. Importantly, these rules are to be applied directly to natural logic sentences and terms, rather than to their would-be predicate logical translations.

Such rules enable deductive querying of the knowledge base giving answer results in the form of classes and more generally compound terms. In principle deductive querying is achievable by introduction of variables ranging over the terms in the natural logic as the metalogical variable X the sample query form

$$\mathsf{X\ isa\ cell\ that\ produce\ hormone}$$

supposed to give the answer betacell. These term variables should not be confused with the quantified entity- or individual variables of predicate logic that range over entities in the application domain of discourse. We account formally for these variables by introduction of a metalogic in which NaturaLog becomes embedded.

As metalogic we can choose a "domesticated" form of predicate logic: In [10,11] we suggested Datalog to this end, and [6] gives elaborate description of the metalogic inference engine, succeeding a more compact presentation in [5]. Recall that Datalog consists of definite clauses without compound terms and enjoys decidability.

Let us exemplify the encoding of our natural logic into Datalog: The sentence betacell produce insulin becomes the atomic metalogic clause

$$\mathsf{proposition}(\mathsf{every}, \mathsf{betacell}, \mathsf{produce}, \mathsf{insulin})$$

where the natural logic terms formally appear as, and are treated as, constants.

The sentence betacell isa cell that produce insulin becomes in the metalogic representation

$$\mathsf{proposition}(\mathsf{every}, \mathsf{betacell}, \mathsf{isa}, \mathsf{cell\text{-}that\text{-}produce\text{-}insulin})$$

where cell-that-produce-insulin is a new simple concept term that becomes defined by the following pair of defining metalogic clauses

$$\mathsf{definition}(\mathsf{cell\text{-}that\text{-}produce\text{-}insulin}, \mathsf{isa}, \mathsf{cell})$$
$$\mathsf{definition}(\mathsf{cell\text{-}that\text{-}produce\text{-}insulin}, \mathsf{produce}, \mathsf{insulin})$$

The decomposition applies recursively to nested concept terms.

The distinction between definitional and non-definitional contributions in the decomposition is internal to the system. However, [11] hints at further introducing an external epistemic distinction between definitional and observational sentences in the knowledge base.

3.1 The Encoded Knowledge Base as Graph

The metalogic knowledge base representation may be conceived as a labeled graph whose nodes are concept terms, and whose directed arcs represent relationships with accompanying determiners (quantifiers). The graph picture further supports the conception of the knowledge base as an extended ontology. The concept terms present in the KB sentences are uniquely represented as nodes in the graph across sentences. Moreover the graph view helps visualizing pathway querying, cf. Sect. 5. In the decomposition of sentences into clauses there is no loss of information.

4 Design of Inference Engine

Natural logics use high level inference rules reflecting "intuitive" rules applied by humans when reasoning with descriptions in natural language. This adds to the explainability of the deductive reasoning and hence the query processing. These rules are now to be formalized in the metalogic, exploiting the decomposed and encoded NaturaLog sentences.

As a key principle, answers to queries stated to the knowledge are computed by use of the inference rules. We refer to [5,6] for the rules we apply for NaturaLog. These papers also contain references to the background literature dealing with deductive reasoning in natural logics.

We present here using Datalog only a few rules. As key rules there are the so-called monotonicity rules:

$\mathsf{proposition}(\mathsf{every}, C, R, Dsuper) \leftarrow$
$\qquad \mathsf{proposition}(\mathsf{every}, C, R, D) \wedge \mathsf{proposition}(\mathsf{every}, D, \mathsf{isa}, Dsuper)$
$\mathsf{proposition}(\mathsf{every}, Csub, R, D) \leftarrow$
$\qquad \mathsf{proposition}(\mathsf{every}, C, R, D) \wedge \mathsf{proposition}(\mathsf{every}, Csub, \mathsf{isa}, C)$

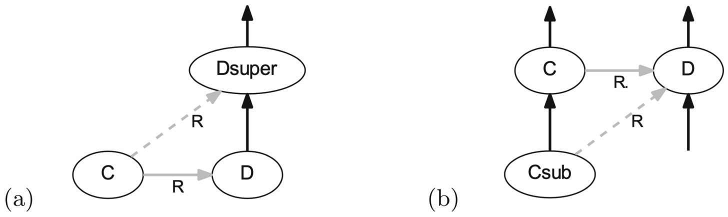

Fig. 1. Monotonicity rules: (a) inheritance and (b) generalization. Dashed relations are inferred.

The graphs for these are shown in Fig. 1. One observes that the latter rule provides "property inheritance". Further, one may observe that the transitivity rule for isa obtains with the special case of R being instantiated to isa. As another distinctive feature we introduce a rule for obtaining the corresponding passive voice sentence from a given active voice sentence. This implies in particular that the sentence form C isa D gives rise to some D isa C as mentioned in Sect. 2.3 besides giving the weakened some C isa D by means of appropriate rules. In Sect. 5.1 we mention the potential for extension with "non-logical" application-specific inference rules.

4.1 Materialization of Deductive Closure

In [7] we propose that the part of the deductive closure of a knowledge base being relevant for query answers of a knowledge base is computed and stored in advance in a compilation process jointly involving the sentences. This means that sentences and terms potentially appearing in a query answer is made present in advance in the compiled knowledge base.

Furthermore, in [7] (forthcoming 2020) we elaborate a version of the inference engine where DATALOG is replaced by relational database query operations. This enables use of a database system for efficient retrieval of sentences in the knowledge base, and in addition inference computations are made algorithmically more efficient by "bulk processing" applying database join operations.

The recursive NATURALOG syntax generally admits infinitely many terms. However, only a finite subset of these are known to have a non-empty denotation in the form of subconcepts in the knowledge base – namely either by their being explicitly present or by being a superclass of such a mentioned term.

5 Systems Functionalities

A range of systems functionalities can be obtained on basis of the relevant deductive closure computed by the inference rules. First of all there are answers in the form of sets of terms from instantiation of metalogical variables in query forms exemplified by

$$\mathsf{proposition}(\mathsf{every},\ X, \mathsf{produce}, \mathsf{hormone})$$

supposed to give as answer cells that produce hormone, such as betacells. Notice here that computation of such answers in general draws on inference rules, say, for combining the sentence betacell produce insulin with the sentence insulin isa hormone in a monotonicity inference rule. Query answer terms may well be compound terms stemming from a given sentence or having been computed in the compilation process.

So far, we accept only affirmative sentences in a knowledge base. Negative sentences may be accepted as query sentences in the form no $C \quad R \quad D$ being logically contrary to every $C \quad R \quad D$ and contradictory to some $C \quad R \quad D$, with a supporting inference rule appealing to negation by non-provability.

The graph conceptualization of a NATURALOG knowledge base as (usually) one coherent graph invites path-finding operations for retrieving shortest paths between two given terms as discussed and examplified in [1,3,4]. Path-finding is of particular interest for tracing pathways in life-science knowledge bases.

5.1 Application-Specific Query Inference Rules

In our two level logic setup the natural logic level describes the application domain of discourse and the metalogic level prescribes the computing with natural logic sentences and terms. This opens for introducing application specific rules in the metalogic. For instance, one easily introduces a rule providing a verb, say, "causes" with the property of transitivity of the underlying relation.

As another example, [6] describes an additional metalogic rule for computing the commonalities of two given terms. When asking for instance about common properties of the two concepts alphacell and betacell, the deduced answer may comprise informative compound terms such as cell that produce hormone. More sophisticated general rules may afford the computing of analogies, asking for instance which concept is related to alphacell as insulin is related to betacell.

Inference rules may be introduced to verify ad hoc consistency requirements formulated as rules expected to yield empty query answers in case of consistency fulfilment as known from logic programming.

6 Conclusion and Open Problems

The described natural logic with the accompanying realization principles attempts to strike a balance between on one hand language expressivity and interesting computational functionalities and on the other hand an acceptable computational tractability. NaturaLog covers basic essential application domain demands within the considered life-science domains, but additional useful features are to be included in coming versions. Among the possible semantical extensions let us just mention exception handling for non-monotonic blocking of unrestricted inheritance of properties, and introduction of generalized quantifiers such as "most" and "few".

It remains to be verified that computational tractability can be obtained with the suggested relational database implementation, when scaling up to interesting large size knowledge bases.

An interesting but highly challenging problem is to conduct a computer-assisted, if not completely automatic, translation of essential parts of given natural language descriptive texts into NaturaLog. This complex and difficult problem of computationally extracting natural logic sentences from descriptions in free natural language is touched in [1,3,4]. To this end a syntactic-semantic analysis using NaturaLog as target language might be ameliorated by inductive machine learning methods. Eventually, more expressive versions of natural logic may come into use directly as logical specification languages in natural science domains.

References

1. Andreasen, T., Bulskov, H., Nilsson, J.F., Jensen, P.A.: Computing pathways in bio-models derived from bio-science text sources. In: Proceedings of the IWBBIO International Work-Conference on Bioinformatics and Biomedical Engineering, Granada, April, pp. 217–226 (2014)
2. Andreasen, T., Bulskov, H., Nilsson, J.F., Jensen, P.A.: A system for conceptual pathway finding and deductive querying. Flexible Query Answering Systems 2015. AISC, vol. 400, pp. 461–472. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-26154-6_35
3. Andreasen, T., Bulskov, H., Nilsson, J.F., Jensen, P.A.: Partiality, Underspecification, and Natural Language Processing, chapter A Natural Logic for Natural-Language Knowledge Bases. Cambridge Scholars, Newcastle upon Tyne (2017)

4. Andreasen, T., Bulskov, H., Jensen, P.A., Nilsson, J.F.: Pathway computation in models derived from bio-science text sources. In: Kryszkiewicz, M., Appice, A., Ślęzak, D., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2017. LNCS (LNAI), vol. 10352, pp. 424–434. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60438-1_42
5. Andreasen, T., Bulskov, H., Jensen, P.A., Nilsson, J.F.: Deductive querying of natural logic bases. In: Cuzzocrea, A., Greco, S., Larsen, H.L., Saccà, D., Andreasen, T., Christiansen, H. (eds.) FQAS 2019. LNCS (LNAI), vol. 11529, pp. 231–241. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-27629-4_22
6. Andreasen, T., Bulskov, H., Nilsson, J.F., Jensen, P.A.: Natural logic knowledge bases and their graph form, p. 34. (2020). Submitted to journal (under review)
7. Andreasen, T., Bulskov, H., Nilsson, J.F.: A natural logic system for large knowledge bases. In: The 30th International Conference on Information Modelling and Knowledge Bases, Ejc 2020, Hamburg, Germany, 8–12 June 2020 (2020)
8. Klima, G.: Natural logic, medieval logic and formal semantics. Magyar Filózfiai Szemle **54**(4), 58–75 (2010)
9. Moss, L.S.: Syllogistic logics with verbs. J. Log. Comput. **20**(4), 947–967 (2010)
10. Fischer Nilsson, J.: Querying class-relationship logic in a metalogic framework. In: Christiansen, H., De Tré, G., Yazici, A., Zadrozny, S., Andreasen, T., Larsen, H.L. (eds.) FQAS 2011. LNCS (LNAI), vol. 7022, pp. 96–107. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24764-4_9
11. Nilsson, J.F.: In pursuit of natural logics for ontology-structured knowledge bases. In: The Seventh International Conference on Advanced Cognitive Technologies and Applications (2015)
12. van Benthem, J.: Essays in logical semantics. In: Studies in Linguistics and Philosophy, vol. 29. D. Reidel, Dordrecht (1986)