Computational Aspects of Constraint-based Linguistic Description III

Jochen Dörre
(editor)

DYANA-2

Dynamic Interpretation of Natural Language
ESPRIT Basic Research Project 6852
Deliverable R1.2.C
September 1995

Computational Aspects of Constraint-based Linguistic Description III

Jochen Dörre
Universität Stuttgart
(editor)

authors:
Mark Johnson and Jochen Dörre
Jochen Dörre and Suresh Manandhar

DYANA-2

Dynamic Interpretation of Natural Language
ESPRIT Basic Research Project 6852

Deliverable R1.2.C
September 1995

ESPRIT Basic Research Project 6852 DYANA-2

Dynamic Interpretation of Natural Language

Universiteit van Amsterdam
Institute for Logic, Language and Computation (ILLC)

University of Edinburgh
Centre for Cognitive Science (ECCS)

Universität München
Centrum für Informations- und Sprachforschung (CIS)

Universitetet i Oslo
Department of Linguistics and Philosophy (ILF)

Universität Stuttgart
Institut für Maschinelle Sprachverarbeitung (IMS)

Universität Tübingen
Seminar für Sprachwissenschaft (SfS)

Universiteit Utrecht
Research Institute for Language and Speech (OTS)

For copies of reports, updates on project activities and other DYANA-related information, contact:

The DYANA-2 Project Administrator
ILLC/Department of Philosophy
University of Amsterdam
Nieuwe Doelenstraat 15
NL-1012 CP Amsterdam

http://www.fwi.uva.nl/research/illc/dyana/Home.html

Contents

Introduction

The present deliverable R1.2.C contains two articles which both report on work done outside DYANA (in terms of funding). We have decided to include these additional contributions in this unplanned deliverable, because they document a direct continuation and reaction to DYANA-funded work in the area of constraint-based grammar formalisms.

The first paper, by Mark Johnson and myself, describes the Lemma Table Proof Procedure (LTPP), a memoisation technique that can handle co-routined constraints in the framework of CLP. It extends previous work by the authors to apply to general CLP languages.[1]

The LTPP is a generalisation of Earley Deduction which terminates on a larger class of (logic) programs. A characteristic feature that deserves special attention is its flexible control rule. With it the programmer can compose and fine-tune the mixture of processing strategies to be used for a given program in a given application. The aspects which admit calibration are

- which goal to memo and which to evaluate volatile;
- which goals to pass as constraints into a subcomputation and which not to pass;
- which goals to regard as residue in a subcomputation and to pass out again;
- and how much to abstract from an initial goal of a memoed subcomputation to increase the chance of reusability of its results.

Notice that the abstraction mentioned in the last point fixes the amount of top-down information to be used in a proof, i.e., it allows to adjust the proof strategy anywhere in the spectrum between pure top-down (goal-driven) and bottom-up (database-driven) processing on a goal-by-goal basis.

On another issue the LTPP is an important prerequisite to implement efficient best-first search strategies for preference-augmented constraint grammars (of the type proposed, for instance, by Eisele in R1.2.B). An implementation of $A*$ search for probabilistic CUF based on the LTPP is currently carried out and will be part of prototype P3.1+.

The second paper, co-authored by Suresh Manandhar, is related to the work by Esther König, Dov Gabbay and myself presented in DYANA deliverable R1.2.B on a combination of Lambek Logic and Feature Logics.[2] The present paper must be seen more as a reaction to than a continuation of the earlier work. In the latter the question was considered of how Lambek formulae over feature terms could be integrated so that their logic would be in accord with the well-known proof system of unification-based categorial grammar. Somewhat surprisingly this interpretation needs to be rather non-standard. Just from the opposite direction the question of the combination is approached in the current paper: what is the logic of Lambek formulae over feature terms, if we adopt the 'most natural', intuitive semantics?

By the most natural semantics the following is meant. Consider an ordinary Lambek formula (or type) A/B, where A and B are basic types. When

1. cf. [Joh93] and [Dör93]
2. A revised version is to appear [DKG95].

formalising the denotation of such a type, it is standardly assumed that basic types denote fixed *sets* of elements in a given (algebraic) structure (in which an associative 'concatenation' operation $\circ$ is defined[3]). Now without even regarding how A/B is defined, there is a natural way to interpret Lambek formulae where in the place of basic types feature terms (like `cat:np & number:pl`) can appear. We may simply view the standard interpretations of feature terms, which are sets of objects of (first-order) feature structures, as if they were the denotations of basic types! I.e., we can leave unchanged the defining clauses of the semantics of the Lambek operators and the different forms of feature terms and simply throw them together. Only the structure, with respect to which these formulae receive their denotation now needs to contain both the feature relations and the concatenation operation.

The essential new dimension that comes into play when employing feature terms instead of the usual basic types is that they are partially ordered by the subsumption order. Interestingly, in a Lambek system with these new partially ordered 'basic types' an inference $A/B\ B' \Rightarrow A$ can be drawn iff B' is subsumed by (i.e., is a subtype of) B. Hence, application is asymmetric w.r.t. the argument requirement of the functor and the actual argument type. This stands in contrast to unification-based categorial grammar, where B and B' would need to be unified. But is this asymmetry not an obstacle when writing grammars? The contrary seems to be the case. In a recent paper Sam Bayer and Mark Johnson showed that agreement in coordinations, a linguistic issue considered highly problematic for the unification-based approaches, can straightforwardly be modeled correctly in feature-based Lambek logic of the kind described here.

The present paper works out the logic of Lambek formulae over preordered basic types by presenting a complete proof system — a nearly trivial extension to an ordinary Lambek system — and shows that it is generatively equivalent to original Lambek calculus.

Stuttgart, September 1995
Jochen Dörre

References

[DKG95] Jochen Dörre, Esther König, and Dov Gabbay. Fibred semantics for feature-based grammar logic. *Journal of Logic, Language and Information. Special Issue on Language and Proof Theory*, 1995. to appear.

[Dör93] Jochen Dörre. Generalizing Earley deduction for constraint-based grammars. In Jochen Dörre, editor, *Computational Aspects of Constraint-Based Linguistic Description I, DYANA-2 deliverable R1.2.A.* ESPRIT, Basic Research Project 6852, July 1993.

[Joh93] Mark Johnson. Memoization in constraint logic programming. Presented at the *1st Workshop on Principles and Practice of Constraint Programming*, April 28–30 1993. Newport, Rhode Island.

3. Sometimes, e.g., in Ternary Frame Semantics, $\circ$ need not even be functional, but this is unimportant to our point here.

Additional Contribution

Memoization of Coroutined Constraints

Mark Johnson and Jochen Dörre
(Brown University, Universität Stuttgart)

On Constraint-based Lambek Calculi

Jochen Dörre and Suresh Manandhar
(Universität Stuttgart, University of Edinburgh)