Computational Aspects of Constraint-Based Linguistic Description I

Jochen Dörre
(editor)

DYANA-2

Dynamic Interpretation of Natural Language
ESPRIT Basic Research Project 6852

Deliverable R1.2.A
August 1993

Computational Aspects of Constraint-Based Linguistic Description I

Jochen Dörre
Universität Stuttgart
(editor)

authors:
Jochen Dörre and Michael Dorna
Suresh Manandhar
Chris Brew

DYANA-2

Dynamic Interpretation of Natural Language
ESPRIT Basic Research Project 6852

Deliverable R1.2.A
August 1993

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)

Rijksuniversiteit 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

Contents

Introduction

Constraint-based grammar formalisms have obtained their practical importance because of the declarativity and conciseness of linguistic descriptions therein and their logical semantics, which we — now that it is being well-understood — would call simple. Constraint-based descriptions of all levels of linguistic representation have been proposed in the literature underpinning the relevance of this area of research to the field of computational linguistics. The Comprehensive Unification Formalism (CUF) developed during the first phase of DYANA tries to keep the advantages of this class of formalisms as maxims.

As one would expect, the research reported here focuses on CUF, especially on issues of its implementation, algorithms used therein and conceivable extensions of the system. The deliverable contains two regular contributions and two shorter papers which provide commentaries on the research documented by the first two papers.

The first contribution by Michael Dorna and myself gives a detailed account of the current extent of CUF. Since the end of DYANA-1 a considerable amount of work has gone into efforts to increase the robustness and user-friendliness of the system. The biggest novelty is the type system that has been added to CUF. It allows to type domains and ranges of features as well as arguments of predicates with the types possibly being ordered in a hierarchy. Feature typings can, however, be seen also from the other way round, namely that they allow (elements of) types to be defined only for certain features, which map these types to certain other types. A consequence of the hierarchical ordering of the types is that these so-called feature appropriateness conditions get inherited from types to their subtypes. For instance, if we have two features f and g which are typed $f : t_1 \mapsto t_2$ and $g : t_1' \mapsto t_3$ and t_1' is a subtype of t_1, then both features are appropriate for t_1', i.e. its elements are defined for f *and* g, however elements of type t_1 which are not in t_1' may not be defined for g. Notice that since the respective range types also may be involved in appropriateness conditions, it makes sense to talk about appropriateness of feature paths, or more generally, types can be characterized as certain sets of feature structures.

Unlike in other formalisms which allow types to be defined as arbitrary feature terms, type checking in CUF is decidable. This enables us to enforce a rigorous type discipline with the compiler that helps to detect inconsistencies in the program very early. Besides this aspect of error detection there is an interesting computational aspect of the CUF type system. The type hierarchy is stated axiomatically using arbitrary boolean expressions and the component responsible for type consistency, actually a propositional theorem prover, is built into the unification procedure. This lets types behave rather different from sort predicates as concerns disjunctive information. Type consistency is checked very eagerly whereas sorts are handled by the resolution component and can be delayed. Another advantage of CUF's typing declarations that is discussed in the paper is their remarkably simple logical semantics.

The second paper describes more elaborated resolution strategies that proved to be needed when experimenting with memoization-based strategies in the resolution component of the system. Unfortunately some basic assumptions in Earley deduction stand in conflict with requirements of techniques of constraint processing such as goal delaying or co-routining. The problems can however be overcome when using a generalized version of Earley deduction, thus making it possible to apply ideas from chart parsing to the general processing of constraint-based descriptions. An important feature of the proposed method is that it is highly adaptable to the specific needs of the description by the use of control statements.

The paper by Suresh Manandhar, entitled "CUF in context", compares on an informal level the CUF system described in the first paper with the TFS system of Martin Emele and Rémi Zajac on one hand and with the ALE system of Bob Carpenter on the other hand. Although all three systems have a lot of commonalities, foremost the ability to state relations over typed feature structures, they are rather different in the particular choice they make in trading expressivity against efficiency.

Last, but not least, the contribution by Chris Brew addresses the exciting and still difficult question of how declarative formalisms and stochastic methods for language processing can be brought together in a way which allows us to exploit the advantages of both. This paper is naturally of a more speculative nature and was initially meant to only comment on the current state of the art in declarative formalisms with respect to their compatibility to stochastic linguistic processing rather than providing complete solutions. But as it has developed now, I can say that it contains rather concrete suggestions of how to approach this problem. At least, it shows up a promising direction for future research on this topic.

Stuttgart, July 1993
Jochen Dörre

Task 1.2, subtask 1

Computational Aspects of Constraint-based Linguistic Description

CUF – A Formalism for Linguistic Knowledge Representation

Jochen Dörre and Michael Dorna

(Stuttgart University)

Task 1.2, subtask 1

Computational Aspects of Constraint-based Linguistic Description

Generalizing Earley Deduction for Constraint-Based Grammars

Jochen Dörre

(Stuttgart University)

CUF in Context

Suresh Manandhar

(Edinburgh University)

Adding preferences to CUF

Chris Brew

(Stuttgart University)