Lexical Object Theory: Specification Level

JOSÉ F. QUESADA
CICA (Centro Informático Científico de Andalucía), Seville, Spain. E-mail: josefran@cica.es

Abstract. Unification has become a major paradigm in Mathematical and Computational Linguistics. The research done in this area may be classified in four main streams: feature structures as an adequate model for the description of linguistic phenomena, typed unification, representation of feature structures, and unification algorithms. This work proposes a new approach to unification-based Mathematical and Computational Linguistics: the Lexical Object Theory. The main design criteria are based on linguistic motivation, computational efficiency and formal soundness. The first part of the work outlines the main characteristics of the Lexical Object Theory, its comprehensive orientation, and its layered structure based on the separation of the following levels: specification, transformation, typification, representation and unification. The second part concentrates on the specification level of the Lexical Object Theory. The linguistic motivation of this model is presented, as well as a detailed description of the specification formalism, the computational model it is based on, and finally, the inference rules on lexical objects at the specification level.

Key words: computational representation and inference, lexical object theory, lexical object specification formalism, unification-based grammar formalisms

1. Unification-Oriented Mathematical and Computational Linguistics

During the last few decades, unification has become a major paradigm (Kuhn, 1962) in Mathematical and Computational Linguistics. Unification-based grammar formalisms and theories share a common philosophy of work, a kernel of formal definitions, several layers of linguistic, mathematical and computational theories and techniques, and finally, a surface of facts (formal consequences, applied linguistics, computational results, etc.) that the paradigm has to take into account by making use of the criteria of coherence, reliability and efficiency.

Since Joan Bresnan's work on lexically oriented non-transformational linguistics, Ronald Kaplan's Augmented Transition Networks, and Martin Kay's Functional Grammar onwards, there has been an increasing use of the *unification* procedure (as a basic operation for manipulating linguistic information) and *complex feature structures* (as a formal basis for specifying and representing linguistic information).

The use of feature structures has often appeared along the history of grammar theories and formalisms: LFG (Bresnan, 1982), GPSG (Gazdar et al., 1985), DCG (Pereira and Warren, 1980), FUG (Kay, 1985), PATR–II (Shieber, 1984), HPSG (Pollard and Sag, 1994), etc.

A more detailed analysis of the field will allow us to classify the work done within it into four major streams:

Feature structures as an adequate model for the description of linguistic phenomena: The interest for feature structures as an adequate model for the description of linguistic phenomena has motivated the study of their expressive power. As a result of this, several authors have proposed different extensions to the basic model, such as: templates and lexical rules (Shieber, 1984), negation and disjunction (Karttunen, 1984), non-local values and coreference (Kasper and Rounds, 1986), etc.

In the same way, other works apply this theoretical framework in order to find a solution to particular linguistic problems, among which we find sub-categorization (Shieber, 1986), non-local dependencies analysis (Kaplan and Zaenen, 1988) or coordination (Kaplan and Maxwell, 1988).

From a formal point of view, feature structures have some of the following characteristics in common with object-oriented or frame-based knowledge representation languages: expressive power, flexibility, representational adequacy and recursivity. In the same way, feature structures share properties such as having a high level of abstraction, allowing for the definition of classes or types, and incorporating multiple inheritance mechanisms with object-oriented formalisms (Smolka Aït-Kaci, 1990). These properties show the adequacy of the feature structure-based model for the description of lexical and syntactic hierarchies.

The expressive capacity of feature structures is equal to that shown by first order terms in Prolog (Mellish, 1992). However, the flexibility and representational adequacy of feature structures prove them to be more adequate for the description of linguistic phenomena. For this reason, some computational tools have been designed to permit the translation of feature structures into Prolog terms (Schöter, 1993). For example: CLE (Alshawi, 1991), ALEP (Alshawi et al., 1991) and ProFIT (Erbach, 1995).

Typed unification: More recently, some formalisms based on feature structures have incorporated typification mechanisms, obtaining the so-called *sorted (typed) feature structures* (Carpenter, 1992; Messeger et al., 1990). Some of these formalisms are: TDL (Krieger and Schäfer, 1994), ALE (Carpenter and Penn, 1994), CUF (Dörre and Dorna, 1993) and TFS (Emele and Zajac, 1990; Emele, 1994). In this context, and within logic programming, some languages (Life (Aït-Kaci and Lincoln, 1989) and Oz (Smolka et al., 1995)) use typed terms.

Representation of feature structures: The interest for feature structures goes beyond the mere description of linguistic phenomena. Other works study the *representation of feature structures*. Specifically, they study the denotational semantics of the model: Shieber, (1984); Pereira and Shieber, (1984); the computationally most efficient data structures which permit the representation of this information: