Unification Categorial Grammar:
A Concise, Extendable Grammar for Natural Language Processing

Jonathan CALDER, Ewan KLEIN and †Henk ZEEVAT

University of Edinburgh
Centre for Cognitive Science
2 Buccleuch Place
Edinburgh
EH8 9LW

†Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
Keplerstrasse 17
D 7000 Stuttgart

Abstract Unification Categorial Grammar (UCG) combines the syntactic insights of Categorial Grammar with the semantic insights of Discourse Representation Theory. The addition of unification to these two frameworks allows a simple account of interaction between different linguistic levels within a constraining, monostratal theory. The resulting, computationally efficient, system provides an explicit formal framework for linguistic description, within which large fragments of grammars for French and English have already been developed. We present the formal basis of UCG, with independent definitions of well-formedness for syntactic and semantic dimensions. We will also focus on the concept of *modifier* within the theory.

1. Introduction

Unification Categorial Grammar (UCG) combines the syntactic insights of Categorial Grammar with the semantic insights of Discourse Representation Theory (DRT, Kamp 1981). The addition of unification (Shieber et al. 1983) to these two frameworks allows a simple account of interaction between different linguistic levels. The resulting, computationally efficient, system provides an explicit formal framework for linguistic description, within which large grammar fragments for French (Baschung et al. 1987) and English (Calder, Moens and Zeevat 1986) have already been developed. This paper will describe the design of the UCG formalism, illustrated by examples of grammatical categories and rules.[1]

UCG embodies several recent trends in linguistics. First, being a categorial grammar, it is strongly lexicalist. In other words relatively little information is contained in grammar rules. Most information originates in the lexicon. Second, it is strictly declarative. Unification is the only operation allowed over grammatical objects. Third, there is a very close relationship between the syntax and semantics of linguistic expressions.

UCG lies within the family of grammars described by Uszkoreit 1986 and Karttunen 1986. UCG also has close affinities to the Head-Driven Phrase Structure Grammar (HPSG) proposed by Pollard 1985. The main theoretical difference is that in HPSG well-formedness is characterized algorithmically, rather than declaratively as in UCG. For this reason, we have adopted Pollard's terminology and refer to linguistic expressions as *signs*. A sign represents a complex of phonological, syntactic and semantic information, each of these linguistic levels having its own definitions of well-formedness.

In UCG, we employ three primitive categories: nouns (**noun**), sentences (**sent**) and noun phrases (**np**) These primitive categories admit further specification by features, so that we can distinguish finite and non-finite sentences, nominative and accusative NPs, and so on. Categories are now defined as follows:

(1) a.
Any primitive category (together with a syntactic feature specification) is a category.

b.
If A is a category, and B is a sign, then A/B is a category.

In a category of the form **A/B**, we call **B** the *active* part of the category, and also of the sign as a whole in which **A/B** occurs as category. It will be observed that this definition is just the categorial analog of Pollard's (1985) proposal for subcategorization, according to which phrasal heads are specified for a list of signs corresponding to their complements. Likewise, (1) is closely related to the standard definition for the set of categories of a categorial grammar.

Within the grammar, we allow not just constant symbols like **sent** and **np**, but also variables, at each level of representation. Variables allow us to capture the notion of incomplete information, and a sign which contains variables can be further specified by unification. The form of unification that we rely on is first-order term unification, provided as a basic operation in programming languages such as PROLOG.

This, in essence, is the structure of UCG. We will complicate the picture by distinguishing two rules of functional application, and by giving more content to the notions of semantics and features.

2. The Mechanisms of UCG

2.1. Structuring Signs

UCG signs have three basic components, corresponding to their phonology, category and semantics. We will write the most unspecified sign as follows:

(2) W:
C:
S

by which we intend a sign with phonology **W**, category **C** and semantics **S**. (1) and (2) give well-formedness conditions on possible instantiations for a sign's category. For the present paper, we will assume that a sign's phonology may be simply its orthography in the case of proper names, otherwise a sign's phonology may be composite, consisting of variables and orthographic constants separated by +. The + operator is understood as denoting concatenation.[2]

[1] The work described here is supported by the EEC ESPRIT projec P393 *ACORD*: the Construction and Interrogation of Knowledge Bases usin Natural Language Text and Graphics.

[2] This operation might appear to take us beyond the bounds of first-order unification. However in the cases we will deal with, there are equivalent signs which express concatenation by means of PROLOG difference lists.

2.2. Indexed Language

The semantic representation language that we use to encode the semantics of a sign is called InL (for Indexed Language), and is derived from Discourse Representation Theory (cf. Kamp 1981), supplemented with a Davidsonian treatment of verb semantics (cf. Davidson 1967). The main similarity with the Discourse Representation languages lies in the algebraic structure of InL. There are only two connectives for building complex formulas; an implication that at the same time introduces universal quantification, and a conjunction.

The language InL differs in one important respect from the DRT formalism, and thus earns its name; every formula introduces a designated variable called its **index**. This does not mean that (sub)formulas may not introduce other variables, only that the index has a special status. The postulation of indices is crucial for the treatment of modifying expressions, but it is independently plausible on other grounds. Every sign has an associated ontological type, represented by the *sort* of its index. Subsumption relations hold between certain sorts; for instance, a index of sort **singular** will unify with an index of sort **object** to yield an index of sort **singular**. For notational purposes, we use lower case alphabetics to represent sorted variables in InL formulas. Upper case alphabetics are variables over formulas. The index of an expression appears within square brackets in prenex position. (3) gives example translations of some expressions.

(3)	*[Index]*	*Formula*	*Expression*	*Sort*
a.	[e]	WALK(e, x)	walk	event
b.	[x]	STUDENT(x)	student	singular object
c.	[x]	[PARK(y),[x][IN(x,y),MAN(x)]]	man in a park	singular object
d.	[m]	BUTTER(m)	butter	mass object
e.	[s]	STAY(s, x)	stay	state

3. UCG Binary Rules

We may write UCG rules as simple relations between signs. We require two rules, our analogs of forwards and backwards application (FA and BA respectively). Here we follow the PROLOG convention that variables start with an upper case alphabetic.

(4) (FA) Phonology:Category/Active:Semantics Active
↓
Phonology:Category:Semantics

(BA) Active Phonology:Category/Active:Semantics
↓
Phonology:Category:Semantics

These rules state that in the case of function application, the resulting category is simply that of the functor with its active sign removed; the semantics and phonology of the result are those of the functor, thus effecting a very strict kind of Head Feature Convention. Note in particular that we view the phonological, syntactic and semantic functor as *always* coinciding. This has important consequences for the way we treat quantified NP's, as we discuss in the next section. Any of the features of the resulting sign may have been further instantiated in the process of unification.

Importantly, (4) states that a functor may place restrictions on any of the dimensions of its argument. Likewise it will determine the role that the information expressed by its argument plays in the resulting expression. A UCG sign thus represents a complex of constraints along several dimensions.

4. UCG Signs

We now give some example UCG signs.

4.1. Nouns and Adjectives

(5) student:
noun:
[x]STUDENT(x)

(6) cheerful+W:
noun/(W:noun:[x]P):
[x][CHEERFUL(x), [x]P]

The reader is invited to work out for herself how the signs (5) and (6) will combine using the rule of forwards application.

4.2. Determiners

Following Montague 1973, we treat quantified NPs as type-raised terms. We can however take advantage of the polymorphic nature of UCG categories and have a single representation for NPs regardless of their syntactic context. In our analysis, the determiner introduces the type raising. This is the sign that corresponds to *a*.

(7) W
(C/(W:C/ a+W1: np[nom or obj]:b):[a]S)/(W1:noun:[b]R):
[a][[b]R, S]

More verbosely, this says that *a* combines first with a noun which has phonology **W1** and semantic index **b**. The semantics that results from such a combination is a conjunction, the first conjunct of which is the semantics of the noun. The second conjunct is the semantics of the resulting NP's argument. As the NP is type-raised, it has a category of the schematic form:

(8) C/(C/np)

That is, a type-raised NP will take as its argument some constituent which was itself to combine with a (non-type-raised) NP. When fleshed out with values for the other components of a sign, it will have the form as shown in (9). Note in particular that, as it is the verb that determines linear order, the phonology of the resulting expression depends on that of the argument to the NP. (9) shows the result of combining (7) and (5) via forward application.

(9) W
C/(W:C/ a+student: np[nom or obj]:b):[a]S:
[a][[b]STUDENT(b), S]

The sign corresponding to *every* (10) is very similar to that for *a*, the major difference being that *every* introduces DRT implication, notated here with ⇒.

(10) W
(C/(W:C/ every+W1: np[nom or obj]:b):[a]S)/(W1:noun:[b]R):
[s][[b]R ⇒ [a]S]

4.3. Verbs

The following is the sign for *walks*:

(11) W+walks:
sent[fin]/(W:np[nom]:x):
[e]WALK(e, x)

This will combine with the sign (9) *a student* to yield (12) of which the semantics may be read as: "There is a walking event, of which b is the agent, and b is a student".

(12) a+student+walks:
sent[fin]:
[e][[b]STUDENT(b), WALKS(e, b)]

5. Modifiers in UCG

We have already seen one category of modifiers, namely adjectives, in section 4.1. We are able to make more general statements about modifiers; they all contain instances of the category in (13):

(13) C/(W:C:S)

Appropriate restrictions on C will allow us to describe, for instance, the class of VP modifiers such as adverbials and auxiliaries (Cf. Bouma 1988). The close relationship between syntax and semantics allow us to give concise formulations of the distinctions between *intersective*, *vague* and *intensional* modifiers (Kamp 1975). In the first two cases, the semantics of the modified expression is conjoined with that of the modifying expression. In the vague case, we have to relativize the meaning of the modifying expression to that of the modified. In the third case, the semantics of the modified expression must be contained within the scope of an intensional predicate. The following examples illustrate the three cases.

(14) square+W:
noun/(W:noun:[a]A):
[a][SQUARE(a), A]

large+W:
noun/(W:noun:[a]A):
[a][LARGE(A, a), A]

fake+W:
noun/(W:noun:[a]A):
[a]FAKE(A)

6. Further development of the theory

We have not attempted to give a fully representative list of UCG signs here. Elsewhere (Calder, Moens and Zeevat 1986, Zeevat, Klein and Calder 1987), substantial analyses of subcategorization, prepositional and adverbial modification, negation, relative clauses and sentential connectives have been developed. We have also extended the theory to encompass non-canonical word order, using a mechanism similar to the GPSG **SLASH** (Gazdar et al. 1985), and to handle quantificational constraints on anaphora following the analysis of Johnson and Klein 1986. We have an efficient implementation of the system which represents signs as PROLOG terms, using a different treatment of phonological information. The use of templates (Shieber et al. 1983) allows us to capture generaliztions about clases of lexical items. The compilation of UCG structures into PROLOG terms is performed by a general processor driven by the definitions of well-formedness of the dimensions of a sign, allowing compile-time type-checking of grammars (Calder, Moens and Zeevat 1986). The system uses a tabular shift-reduce parser.

7. Further developments of UCG

The system described above is deficient in some respects. For example, requiring coincidence between the phonological, syntactic and semantic functors may be too strict. The problem of quantifier scoping is a case in point. Zeevat 1987 suggests relaxing this requirement to allow linear ordering to become the dominant factor in determining semantic functor-argument relations. It seems likely that such a step will also be necessary for certain phonological phenomena. Current work is investigating the utility of associative and commutative unification in this respect.

In extending UCG to allow treatment of unbounded dependency constructions and partially free word order, heavy use is made of unary rules (Wittenburg 1986). Current work aims to recast the notion of unary rule within the framework of paramodular unification (Siekmann 1984).

8. Conclusion

An attractive feature of UCG is the manner in which different levels of representation - semantic, syntactic and phonological - are built up simultaneously, by the uniform device of unification. There are, of course, different organizing principles at the different levels; conjunction and implication exist at the semantic level, but not at the syntactic or phonological. Nevertheless, the compositional construction of all three levels takes place in the same manner, namely by the accretion of constraints on possible representations. Although we have said nothing substantive about phonology here, it seems plausible, in the light of Bach and Wheeler 1981 and Wheeler 1981, that the methodological principles of compositionality, monotonicity and locality can also lead to illuminating analyses in the domain of sound structure.

UCG is distinctive in the particular theory of semantic representation which it espouses. Two incidental features of InL may obscure its relation to DRT. The first is very minor: our formulas are linear, rather than consisting of "box-ese". The second difference is that we appear to make no distinction between the set of conditions in a DRS, and the set of discourse markers. In fact, this is not the case. A simple recursive definition (similar to that for "free variable" in predicate logic) suffices to construct the cumulative set of discourse markers associated with a complex condition from indices within a formula. This definition also allows us to capture the constraints on anaphora proposed by DRT. These departures from the standard DRT formalism do not adversely affect the insights of Kamp's theory, but do offer a substantial advantage in allowing a rule-by-rule construction of the representations, something which has evaded most other analyses in the literature.

UCG syntax is heavily polymorphic in the sense that the category identity of a function application typically depends on the make-up of the argument. Thus, the result of applying a type-raised NP to a transitive verb phrase is an intransitive verb phrase, while exactly the same functor applied to an intransitive verb phrase will yield a sentence. Analogously, a prepositional modifier applied to a sentence will yield a sentence, while exactly the same functor applied to a noun will yield a noun. This approach allows us to dramatically simplify the set of categories employed by the grammar, while also retaining the fundamental insight of standard categorial grammar, namely that expressions combine as functor and argument. Such a mode of combination treats head-complement relations and head-modifier relations as special cases, and provides an elegant typology of categories that can only be awkwardly mimicked in X-bar syntax.

Finally, we note one important innovation. Standard categorial grammar postulates a functor-argument pair in semantic representation which parallels the syntactic constituents; typically, lambda-abstraction is required to construct the appropriate functor expressions in semantics. By contrast, the introduction of signs to the right of the categorial slash means that we subsume semantic combination within a generalized functional application, and the necessity of constructing specialized functors in the semantics simply disappears.

References

Bach, E. and Wheeler, D. (1981) Montague Phonology: A First Approximation. In Chao, W. and Wheeler, D. (eds.) *University of Massachusetts Occasional Papers in Linguistics*, Volume 7, pp27-45. Distributed by Graduate Linguistics Student Association, University of Massachusetts.

Baschung, K., Bes, G. G., Corluy, A. and Guillotin, T. (1987) Auxiliaries and Clitics in French UCG Grammar. In *Proceedings of the Third Conference of the European Chapter of the Association for Computational Linguistics*, 1987.

Bouma, G. (1988) Modifiers and specifiers in categorial unification grammar. *Linguistics*, **26**, 21-46.

Calder, J., Moens, M. and Zeevat, H. (1986) A UCG Interpreter. ESPRIT PROJECT 393 ACORD; Deliverable T2.6.

Davidson, D. (1967) The logical form of action sentences. In Rescher, N. (ed.) *The Logic of Decision and Action*. Pittsburgh: University of Pittsburgh Press.

Gazdar, G., Klein, E., Pullum, G. and Sag, I. (1985) *Generalized Phrase Structure Grammar*. London: Basil Blackwell.

Johnson, M. and Klein, E. (1986) Discourse, anaphora and parsing. In *Proceedings of the 11th International Conference on Computational Linguistics and the 24th Annual Meeting of the Association for Computational Linguistics*, Institut fuer Kommunikationsforschung und Phonetik, Bonn University, Bonn, August, 1986, pp669-675.

Kamp, J. A. W. (1975) Two Theories about Adjectives. In Keenan, E. L. (ed.) *Formal Semantics of Natural Language: Papers from a colloquium sponsored by King's College Research Centre, Cambridge*, pp123-155. Cambridge: Cambridge University Press.

Kamp, H. (1981) A theory of truth and semantic representation. In Groenendijk, J. A. G., Janssen, T. M. V. and Stokhof, M. B. J. (eds.) *Formal Methods in the Study of Language*, Volume 136, pp277-322. Amsterdam: Mathematical Centre Tracts.

Karttunen, L. (1986) Radical Lexicalism. Report No. CSLI-86-68, Center for the Study of Language and Information, December, 1986. Paper presented at the Conference on Alternative Conceptions of Phrase Structure, July 1986, New York.

Montague, R. (1973) The proper treatment of quantification in ordinary English. In Hintikka, J., Moravcsik, J. M. E. and Suppes, P. (eds.) *Approaches to Natural Language*. Dordrecht: D. Reidel. Reprinted in R H Thomason (ed.) (1974), *Formal Philosophy: Selected Papers of Richard Montague*, pp247-270. Yale University Press: New Haven, Conn.

Pollard, C. J. (1985) Lectures on HPSG. Unpublished lecture notes, CSLI, Stanford University.

Shieber, S., Uszkoreit, H., Pereira, F. C. N., Robinson, J. J. and Tyson, M. (1983) The Formalism and Implementation of PATR-II. In Grosz, B. and Stickel, M. E. (eds.) *Research on Interactive Acquisition and Use of Knowledge*, SRI International, Menlo Park, 1983, pp39-79.

Siekmann, J. H. (1984) Universal Unification. In Shostak, R. H. (ed.) *Proceedings of the Seventh International Conference on Automated Deduction*, Napa, California, May, 1984, pp1-42. Lecture Notes in Computer Science, Springer-Verlag.

Uszkoreit, H. (1986) Categorial Unification Grammars. In *Proceedings of the 11th International Conference on Computational Linguistics and the 24th Annual Meeting of the Association for Computational Linguistics*, Institut fuer Kommunikationsforschung und Phonetik, Bonn University, Bonn, 25-29 August, 1986, pp187-194. Also Center for the Study of Language and Information.

Wheeler, D. (1981) Aspects of a Categorial Theory of Phonology. PhD Thesis, Linguistics, University of Massachusetts at Amherst. Distributed by Graduate Linguistics Student Association, University of Massachusetts.

Wittenburg, K. W. (1986) Natural Language Parsing with Combinatory Categorial Grammar in a Graph-Unification-Based Formalism. PhD Thesis, Department of Linguistics, University of Texas.

Zeevat, H., Klein, E. and Calder, J. (1987) An Introduction to Unification Categorial Grammar. In Haddock, N. J., Klein, E. and Morrill, G. (eds.) *Edinburgh Working Papers in Cognitive Science*, Volume 1: *Categorial Grammar, Unification Grammar, and Parsing*.

Zeevat, H. (1987) Quantifier Scope in Monostratal Grammars. Unpublished ms. University of Stuttgart.