# Type Logical Grammar: Categorial Logic of Signs

**Glyn V. Morrill**
(Polytechnic University of Catalonia)

Dordrecht: Kluwer Academic
Publishers, 1994, xiv+307 pp;
hardbound, ISBN 0-7923-3095-1, $99.00,
£66.00, Dfl 165.00

*Reviewed by*
*Fernando Pereira*
*AT&T Labs Research*

Look under the hood of most theories of grammar or computational linguistic formalisms and you will find a "machine," often fueled by "rules," that grinds together (descriptions of) linguistic objects to produce other (descriptions of) linguistic objects. Such machines are justified by their descriptive success, or by claims that they explain (aspects of) linguistic ability. However, the puzzle of the origins of the machine remains. In our present state of knowledge about language, proposed machines can be only indirectly justified. And even if our knowledge were sufficient for more direct justifications, the broader question still remains of what are the *necessary* properties of any sign system that associates an open-ended set of meanings with elements of a **prosodic algebra** of linguistic gestures (spoken, written, signed, ...).[1] Now, that question may just be *too* general. One may argue that language is a contingent product of evolution in all of its interesting aspects. Insofar as there is a biologically distinctive language faculty, its attributes would mostly result from the (co)evolution of mind and language, rather than the instantiation of general principles. But one may also wonder whether there might be ways of characterizing the properties of sign systems while abstracting away from the contingent aspects of human linguistic abilities and practices, in the same way as information theory successfully characterizes the amount of information (loss of uncertainty) conveyable by a channel while abstracting away from contingent aspects of the channel's physical realization and from the use of the information by the recipient. For instance, the finiteness of language processors demands that the meanings of sufficiently complex signs be a function of the meanings of their parts: thus compositionality.

Without additional constraints, compositionality could be trivially satisfied (Zadrozny 1994). But the fact that language users derive their implicit contract for sign meaning from finite evidence imposes strong uniformity requirements on signs. While this argument has not, to my knowledge, been made rigorous through an axiomatic treatment, type-logical grammar offers a promising notion of uniformity: to say, for instance, that a sign $c$ meaning $z$ has type $B/A$ is to say that $c+a$ (where $+$ is a suitable sign combination operator) has type $B$ and means $z(x)$, given that $a$ has type $A$ and means $x$. But uniformity must also go backwards, if the use and meaning of a sign is to be induced from its appearance with other signs of appropriate type. If we observe

1 Morrill says: "We shall henceforth refer to the dimension opposed to semantics as prosodics, reserving the term syntax for the bridge between prosodics and semantics. Prosodics may be word order, phonology or phonetics depending on the detail targeted, and is intended as a quite general term for the symbols of a language, including e.g. the signals in sign language.... We begin by assuming an algebra of prosodic objects and an algebra of semantic objects." (pp. 12–13)

that sign $c$ combines with varying signs $a$ of type $A$ and meaning $x$ to yield $c + a$ of type $B$ and meaning $u$ (depending on $x$), then we infer that $c$ has type $B/A$ and means $\lambda x.u$. These are just two of the rules of the Lambek–van Benthem calculus, in which Lambek's logical approach to categorial grammar and van Benthem's categorial semantics are brought together.

The Lambek–van Benthem use of a logic to specify the link between prosodic and semantic forms via categories and their corresponding types is the simplest instance of the systems that Morrill explores in depth in *Type Logical Grammar*. Although the book's title is (deliberately?) ambiguous, I use the disambiguated term *type-logical grammar* in the more technical discussion of the book that follows to make clear that we are dealing with grammars formalized with type logics, that is, logics that express notions of well-typedness for underlying algebras.

In general, types in type-logical grammar classify signs according to how they can serve as inputs or appear as outputs of prosodic operators. For each $n$-ary prosodic operation we have a **forward** type connective to construct the type of the outputs of the operation for $n$ given input argument types, and conversely $n$ **residuation** type connectives to construct the type of inputs for argument $i$ that, combined with values of given types for the other $n - 1$ arguments, yield outputs of a given type. Type logic describes the inferential relationships between types that arise from the meaning of those types as constraints on the inputs and outputs of prosodic operations. For instance, in the Lambek calculus the inference $B/A, A \Rightarrow B$ holds: any sign of type $B/A$ concatenated with any sign of type $A$ yields a sign of type $B$. On the semantic side, forward type connectives correspond to functional application, while residuation connectives correspond to abstraction over the relevant argument positions. The big payoff of type-logical grammar is that the meaning of a sign combination follows from the meanings of the signs and from the way in which the type of the combination follows from the types of the signs being combined. In other words, the types and type inferences of the type logic characterize exactly the semantic uniformities afforded by a type-logical grammar. Types and allowed type inferences are not arbitrary formal machinery, but instead reflect precisely the combinatory possibilities of the underlying prosodic algebra.

The relationship between type-logical inferences and the corresponding semantic composition recipes, expressed as $\lambda$-terms, is a version of the famous Curry-Howard correspondence (Howard 1980) between intuitionistic formulas and function types, and between intuitionistic proofs and $\lambda$-terms. The pairing between proofs and terms also has a type-checking interpretation: a $\lambda$-term has an associated proof exactly when it is well-typed (Hindley and Seldin 1986). Intuitionistic type logic, however, is too permissive for type-logical grammar. As the Lambek–van Benthem example above shows, the order of the premises in an inference matters: it is not enough to say that a sign has a certain type, but we must also care about *where* it occurs with that type. That is, premises are **resources** to whose arrangement and multiplicity inference must be sensitive. A helpful intuition with respect to resource sensitivity is that each occurrence of a sign in the derivation of a complex sign provides certain prosodic and semantic material to the complex sign, and thus cannot be simply duplicated or discarded:

> We do not want the set $S$ [of semantic representations of a phrase] to contain *all* meaningful expressions ... which can be built up from the elements of $S$, but only those which use each element exactly once. (Klein and Sag 1985, 172)

Machinery such as $\theta$-roles, case assignment, functional coherence and completeness,

and SUBCAT lists in other grammatical frameworks also encodes some aspects of resource sensitivity. However, those devices are justified purely by their mechanistic impact on grammaticality judgments, while logical resource sensitivity is based on a sharpening of the notion of inference independently of linguistic considerations, as is demonstrated by the extraordinary recent development of linear logic (Girard 1987; Troelstra 1992) and other substructural logics (Schroeder-Heister and Dosen 1993).

Type logics, the Curry-Howard correspondence, and resource sensitivity form the foundation for the book. These are not easy ideas to assimilate, and I am afraid that their presentation in Chapter 2 may be too rushed, with some pesky technicalities being glossed over. To start the presentation with a natural-deduction type-assignment system, especially as given by Morrill (no side conditions on occurrences of discharged variables), could well confuse the reader about the connection between the multiplicity of variables in terms and the corresponding contraction and weakening of type subformulas. Either using side conditions in a natural-deduction system, as Hindley and Seldin (1986) do, or subscripting variables according to the corresponding assumption multisets, as Girard, Lafont, and Taylor (1989) do, would have easily avoided this potential for confusion, which is greater for only arising later when resource sensitivity is introduced.

Past the preparatory material, Chapters 3, 4, and 5 develop one of the main achievements of the book, a recasting of Montague grammar into a purely lexicalized form that dispenses with all the awkward and ad hoc machinery of syntactic and translation rules in the original formulations. The mapping from syntactic categories to types, which in Montague was only conventional, is now central to the system. The categorial connectives used to form new types are enriched to reflect not only the standard predicate-argument constructions of categorial grammar but also the discontinuous wrapping and infixation operators that are the prosodic manifestations of relativization (including pied-piping constructions) and quantification. Intensionality, the interactions of which with quantification were a central concern of Montague's work, is captured with modal type constructors that represent the implicit dependency of the meaning of a sign on the intensional context in which it occurs. Morrill's account is a *tour de force*: given the required interpretations of types as appropriate sets of prosodic algebra elements and the semantic effect of each prosodic operator, the rules of inference for the type logic fall out automatically, and it only remains to give the types of lexical items. No construction-specific rules of construction or interpretation are needed.

It is not possible in a short review to do justice to the range of interesting questions and alternatives that are engaged in the pursuit of the main goals of the book. For example, tradeoffs between nonassociative and associative prosodic construction come up repeatedly. One might argue that nonassociative systems are to some extent a copout, in that they impose distinctions in the prosodic algebra that are not always directly observable, and thus seem to serve as a proxy for an additional level of syntactic representation. On the other hand, associative systems are difficult to keep under control, especially when powerful permutation modalities are brought into play. The book does not resolve these questions completely, nor could it, since they continue to be an area of active debate. One wonders, however, if this tension between overrepresentation and lack of control might not be diagnostic of a deeper insufficiency in the current forms of type-logical grammar.

Chapters 6, 7, and 8 may be seen as further experiments in sharpening the distinctions embodied in prosodic algebras and the corresponding type systems. The polymorphic types of Chapter 6 bring very naturally into type-logical grammar the fine-grained distinctions that feature disjunction and coreference provide in constraint-

based grammars. Morrill points out correctly that computational linguistics has often taken unification as a primary operation rather than as an implementation method for existential instantiation, with consequent complications and confusions arising from thinking too concretely about logical grammar formalisms. However, the point is only partly right. While term unification has the stated origin, unification in attribute-value formalisms has different and independent origins in calculi of type subsumption and in dynamic logics, and it is not obvious that his criticism applies to those uses of unification.

Chapters 7 and 8 look at fine-grained distinctions required to model more accurately constraints on order and extraction. Two basic approaches are explored: either start with a fine-grained prosodic algebra and introduce controlled means of collapsing some of the distinctions (local access to associativity or commutativity); or add to a coarser prosodic algebra operators that mark domains of locality. Both have interesting applications to islands, coordinate structure, and parasitic gaps, among other phenomena, but clearly these explorations are just the initial forays into what appears to be rather complicated territory. The permutation modality seems rather unconstrained, and I wonder why we should not always prefer explicit extraction operators.

The contrast between what I would call "mechanistic" and type-logical approaches to grammar is a main programmatic theme of the book. However, the precise distinction is not always clear, not least because of the historical burden of terminology and concerns type-logical grammarians share with their more mechanistic colleagues. Morrill points out that other grammatical frameworks typically rely on covert levels of representation, and on rules that control the interactions between prosodics, meaning, and internal representation. Such rules and representations encode constraints on grammaticality, but they do not have a direct interpretation in terms of relationships between signs alone. Type-logical grammar has rules, but those play only the metatheoretical role of characterizing sign models, and not the constraining role of rules in constraint-based grammars or combinators in combinatory categorial grammar. That is, the primary ingredients of a grammatical theory are the prosodic algebra, and a uniform way of assigning meanings to elements of the prosodic algebra. Through residuation, semantic composition and abstraction fall out from the prosodic algebra. Type logics are simply a formal way of describing the interactions among those ingredients.

A skeptical reader may feel that Morrill's argument is formal rather than substantive. Standard tricks for mapping between machines and logics are well known. While there are deep technical issues with the relative conciseness of machine and logical descriptions, these do not seem to be immediately relevant here. Instead, Morrill argues that all objects of type-logical grammar have direct denotations in the domain of signs, in contrast with systems that depend on uninterpreted rules and representations. But this argument is slippery: if signs are sufficiently enriched—and one may argue that Morrill himself does play that game in some of his systems—then denotations may be constructed for otherwise purely formal representations. Nevertheless, the argument should not be dismissed out of hand. In type-logical grammar, the combinatory and semantic possibilities of a sign are determined by its type alone, which reflects the prosodic operations that may involve the sign. By choosing the operations to reflect observable features of language as much as possible, we reduce the risk that prosodic terms are being used to encode the hidden state of a linguistic machine, and thus keep closer to the spirit of the type-logical program. A further move away from "encoding" in the prosodic algebra, which Morrill alludes to, is to replace prosodic operators by "accessibility" relations—abstract transitions—between prosodic objects (Kurtonina 1995; Moortgat 1996). The categorial connectives are then modalities interpreted

with respect to those accessibility relations, and categories specify constraints on the prosodic transitions that yield the members of the category. This opens, for instance, the possibility of ambiguous prosodic relations, and generally contributes to the view of grammar as informational constraints rather than mechanism.

A main goal of the foregoing observations is to convince the reader that type-logical grammar, and Morrill's book, is not just formal grammar business as usual. The book covers a very interesting range of linguistic phenomena—quantification, relativization, discontinuous constituency, intensionality, coordination, and locality—and their possible type-logical characterizations, and can be used simply as the most comprehensive study to date of the power and subtleties of the type-logical method. However, facts are disputed, analyses come and go, and no single book can account for more than a fraction of important phenomena. A reader who was to concentrate only on the details of linguistic analysis, however interesting, would in my view miss the possibility of asking some fundamental questions that are not often asked, at least since Harris: what can be said abstractly about what a sign system must be like to be learned from finite evidence and used by bounded users to convey meaning consistently within a user community? And if we start asking such questions, traditional divisions in (computational) linguistics, especially the one between formalists and empiricists, lose much force. If we think of prosodic forms as supplying evidence for meanings, the fundamental empirical question is: what constraints are required for such associations to be efficiently learnable from limited evidence and consistently used? Type-logical grammar is an important way of characterizing the object of the learning task, but it does not have anything to say about the roles of frequency and uncertain evidence in learning, production, and interpretation. Conversely, empirical approaches have not had anything to say about uniformities of sign combination, but only about the associations of particular lexical items. Thus, it may be useful to think of type-logical grammar as defining classes of models satisfying certain (minimal) functional requirements, leaving to empirical studies the task of discovering how particular models—basically possible type assignments of signs—are selected on the basis of finite and possibly noisy evidence. A substantive synthesis would probably require radically new notions of category that better accommodate the ambiguous and multifaceted nature of linguistic evidence, and the creativity of their (re)use, but I cannot see how to even start in that direction without appreciating the main ideas of type-logical grammar. The technical baggage is not light, the going is sometimes rough, but acquaintance with this book's main ideas and achievements will be of value to serious computational linguists who are not afraid to look beyond their noses.

**Errata**

The following could be confusing:

- Page 61, line 17: the minimal intuitionistic sequent that requires weakening is not $B \rightarrow A \Rightarrow B$, but rather $B \Rightarrow A \rightarrow B$, with corresponding type assignment the vacuous abstraction $y : B \Rightarrow \lambda x.y : A \rightarrow B$.
- Page 130, equation (83): $iRi$ should be $iRj$.
- Page 236, equation (37): the top formula of the deduction scheme should be $(B \uparrow C) \downarrow A$, as can be easily verified and is required by the derivation in (39).

**Acknowledgments**

I thank Christine Nakatani and Graeme Hirst for their thorough readings of drafts of this review, which caught quite a few obscurities and infelicities and helped sharpen the arguments. I would also like to thank my collaborators Mary Dalrymple, John Lamping, and Vijay Saraswat, as well as John Fry, for many hours of discussion and extensive correspondence on the connections between type-logical grammar, resource-sensitive logics, and other grammatical frameworks; Glyn Morrill, Michael Moortgat, and Richard Oehrle for illuminating conversations over the years that increased my appreciation of type-logical grammar; and FoLLI, Barcelona, and Florence for the trip and the excellent meals in the summer of 1995 during which many of those interactions took place. All errors, obscurities, and idiosyncrasies that remain are, of course, my own doing.

**References**

Girard, Jean-Yves. 1987. Linear logic. *Theoretical Computer Science*, 50:1–102.

Girard, Jean-Yves, Yves Lafont, and Paul Taylor. 1989. *Proofs and Types*. Cambridge Tracts in Theoretical Computer Science, number 7. Cambridge University Press, Cambridge, England.

Hindley, J. Roger and Jonathan P. Seldin. 1986. *Introduction to Combinators and λ-Calculus*. London Mathematical Society Student Texts, volume 1. Cambridge University Press, Cambridge, England.

Howard, W. A. 1980. The formulae-as-types notion of construction. In J. P. Seldin and J. R. Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*. Academic Press, London, England, pages 479–490.

Klein, Ewan and Ivan A. Sag. 1985. Type-driven translation. *Linguistics and Philosophy*, 8:163–201.

Kurtonina, Natasha. 1995. *Frames and Labels. A Modal Analysis of Categorial Inference*. Ph.D. thesis, Research Institute for Language and Speech, University of Utrecht.

Moortgat, Michael. 1996. Multimodal linguistic inference. *Journal of Logic, Language and Information*, 5(3–4):349–385.

Schroeder-Heister, Peter and Kosta Dosen, editors. 1993. *Substructural Logics*. Oxford University Press, Oxford, England.

Troelstra, Anne Sjerp. 1992. *Lectures on Linear Logic*. CSLI Lecture Notes, number 29. Center for the Study of Language and Information, Stanford, California. Distributed by Cambridge University Press.

Zadrozny, Wlodek. 1994. From compositional to systematic semantics. *Linguistics and Philosophy*, 17:329–342.

*Fernando C. N. Pereira* heads the machine learning and information retrieval research department at AT&T Labs. His research in computational linguistics includes work on logic-based grammar formalisms, parsing, semantic interpretation, statistical language models and finite-state transducers in speech recognition. Pereira's address is AT&T Labs Research, A247, 180 Park Avenue, Florham Park, NJ 07932-0971; e-mail: pereira@research.att.com