PROOF FIGURES AND STRUCTURAL OPERATORS FOR CATEGORIAL GRAMMAR*

Guy Barry, Mark Hepple†, Neil Leslie and Glyn Morrill‡
Centre for Cognitive Science, University of Edinburgh
2 Buccleuch Place, Edinburgh EH8 9LW, Scotland
guy@cogsci.ed.ac.uk, mrh@cl.cam.ac.uk,
neil@cogsci.ed.ac.uk, Glyn.Morrill@let.ruu.nl

ABSTRACT

Use of Lambek's (1958) categorial grammar for linguistic work has generally been rather limited. There appear to be two main reasons for this: the notations most commonly used can sometimes obscure the structure of proofs and fail to clearly convey linguistic structure, and the calculus as it stands is apparently not powerful enough to describe many phenomena encountered in natural language.

In this paper we suggest ways of dealing with both these deficiencies. Firstly, we reformulate Lambek's system using proof figures based on the 'natural deduction' notation commonly used for derivations in logic, and discuss some of the related proof-theory. Natural deduction is generally regarded as the most economical and comprehensible system for working on proofs by hand, and we suggest that the same advantages hold for a similar presentation of categorial derivations. Secondly, we introduce devices called *structural modalities*, based on the structural rules found in logic, for the characterization of commutation, iteration and optionality. This permits the description of linguistic phenomena which Lambek's system does not capture with the desired sensitivity and generality.

LAMBEK CATEGORIAL GRAMMAR

PRELIMINARIES

Categorial grammar is an approach to language description in which the combination of expressions is governed not by specific linguistic rules but by general logical inference mechanisms. The point of departure can be seen as Frege's position that there are certain 'complete expressions' which are the primary bearers of meaning, and that the meanings of 'incomplete expressions' (including words) are derivative, being their contribution to the meanings of the expressions in which they occur. We suppose that linguistic objects have (at least) two components, form (syntactic) and meaning (semantic). We refer to sets of such objects as *categories*, which are indexed by *types*, and stipulate that all complete expressions belong to categories indexed by *primitive types*. We then recursively classify incomplete expressions according to the means by which they combine (syntactically and semantically) with other expressions.

In the 'syntactic calculus' of Lambek (1958) (variously known as *Lambek categorial grammar*, *Lambek calculus*, or **L**), expressions are classified by means of a set of *bidirectional types* as defined in (1).

(1) a. If X is a primitive type then X is a type.
b. If X and Y are types then X/Y and $Y\backslash X$ are types.

X/Y (resp. $Y\backslash X$) is the type of incomplete expressions that syntactically combine with a following (resp. preceding) expression of type Y to form an expression of type X, and semantically are functions from meanings of type Y to meanings of type X.

Let us assume complete expressions to be sentences (indexed by the primitive type S), noun phrases (NP), common nouns (N), and non-finite verb phrases (VP). By the above definitions, we may assign types to words as follows:

(2)

John, Mary, Suzy	:= NP
man, paper	:= N
the	:= NP/N
likes, read	:= (NP\S)/NP
quickly	:= (NP\S)\(NP\S)
without	:= ((NP\S)\(NP\S))/VP
understanding	:= VP/NP

We represent the form of a word by printing it in italics, and its meaning by the same word in boldface. For instance, the form of the word "man" will be represented as *man* and its meaning as **man**.

PROOF FIGURES

We shall present the rules of **L** by means of *proof figures*, based on Prawitz' (1965) systems of 'natural deduction'. Natural deduction was developed by Gentzen (1936) to reflect the natural process of mathematical reasoning in which one uses a number of *inference rules* to justify a single proposition, the *conclusion*, on the basis of having justifications of a number of propositions, called *assumptions*. During

*We would like to thank Robin Cooper, Martin Pickering and Pete Whitelock for comments and discussion relating to this work. The authors were respectively supported by SERC Research Studentship 88306971; ESRC Research Studentship C00428722003; ESPRIT Project 393 and Cognitive Science/HCI Research Initiative 89/CS01 and 89/CS25; SERC Postdoctoral Fellowship B/ITF/206.

†Now at University of Cambridge Computer Laboratory, New Museums Site, Pembroke Street, Cambridge CB2 3QG, England.

‡Now at OTS, Trans 10, 3512 JK Utrecht, Netherlands.

a proof one may *temporarily* make a new assumption if one of the rules licenses the subsequent withdrawal of this assumption. The rule is said to *discharge* the assumption. The conclusion is said to *depend* on the undischarged assumptions, which are called the *hypotheses* of the proof.

A proof is usually represented as a tree with the assumptions as leaves and the conclusion at the root. Finding a proof is then seen as the task of filling this tree in, and the inference rules as operations on the partially completed tree. One can write the inference rules out as such operations, but as these are rather unwieldy it is more usual to present the rules in a more compact form as operations from a set of subproofs (the *premises*) to a conclusion, as follows (where $m \geq 1$ and $n \geq 0$):

(3)
$$\frac{\begin{array}{cccccc} \vdots & & \vdots & [Y_1]^i & & [Y_n]^i \\ X_1 & \cdots & X_{m-n} & X_{m-n+1} & \cdots & X_m \end{array}}{Z}\,R^i$$

This states that a proof of Z can be obtained from proofs of $X_1, \ldots, X_m$ by discharging appropriate occurrences of assumptions $Y_1, \ldots, Y_n$. The use of square brackets around an assumption indicates its discharge. R is the name of the rule, and the index i is included to disambiguate proofs, since there may be an uncertainty as to which rule has discharged which assumption.

As propositions are represented by formulas in logic, so linguistic categories are represented by type formulas in L. The left-to-right order of types indicates the order in which the forms of subexpressions are to be concatenated to give a composite expression derived by the proof. Thus we must take note of the order and place of occurrence of the premises of the rules in the proof figures for L. There is also a problem with the presentation of the rules in the compact notation as some of the rules will be written *as if* they had a number of conclusions, as follows:

(4)
$$\frac{\begin{array}{ccc} \vdots & & \vdots \\ X_1 & \cdots & X_m \end{array}}{\begin{array}{ccc} Y_1 & \cdots & Y_n \end{array}}\,R$$

This rule should be seen as a shorthand for:

(5)
$$\frac{\begin{array}{ccc} \vdots & & \vdots \\ X_1 & \cdots & X_m \end{array}}{\begin{array}{ccc} Y_1 & \cdots & Y_n \end{array}}\,R \qquad \begin{array}{ccc} \vdots & & \vdots \\ \vdots & & \vdots \end{array}$$
$$\frac{\qquad\qquad\qquad\qquad\qquad}{Z}$$

If the rules are viewed in this way it will be seen that they do not violate the single conclusion nature of the figures.

As with standard natural deduction, for each connective there is an *elimination* rule which states how a type containing that connective may be consumed, and an *introduction* rule which states how a type containing that connective may be derived. The elimination rule for / states that a proof of type X/Y followed by a proof of type Y yields a proof of type X. Similarly the elimination rule for \ states that a proof of type $Y\backslash X$ preceded by a proof of type Y yields a proof of type X. Using the notation above, we may write these rules as follows:

(6) a.
$$\frac{\begin{array}{cc} \vdots & \vdots \\ X/Y & Y \end{array}}{X}\,/\mathrm{E}$$
b.
$$\frac{\begin{array}{cc} \vdots & \vdots \\ Y & Y\backslash X \end{array}}{X}\,\backslash\mathrm{E}$$

We shall give a semantics for this calculus in the same style as the traditional functional semantics for intuitionistic logic (Troelstra 1969; Howard 1980). In the two rules above, the meaning of the composite expression (of type X) is given by the functional application of the meaning of the *functor* expression (i.e. the one of type X/Y or $Y\backslash X$) to the meaning of the *argument* expression (i.e. the one of type Y). We represent function application by juxtaposition, so that **likes John** means **likes** applied to **John**.

Using the rules /E and \E, we may derive "Mary likes John" as a sentence as follows:

(7)
$$\dfrac{\dfrac{\textbf{Mary}}{\mathrm{NP}} \quad \dfrac{\dfrac{\textbf{likes}}{(\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP}} \quad \dfrac{\textbf{John}}{\mathrm{NP}}\,/\mathrm{E}}{\mathrm{NP}\backslash\mathrm{S}}}{\mathrm{S}}\,\backslash\mathrm{E}$$

The meaning of the sentence is read off the proof by interpreting the /E and \E inferences as function application, giving the following:

(8) **(likes John) Mary**

The introduction rule for / states that where the rightmost assumption in a proof of the type X is of type Y, that assumption may be discharged to give a proof of the type X/Y. Similarly, the introduction rule for \ states that where the leftmost assumption in a proof of the type X is of type Y, that assumption may be discharged to give a proof of the type $Y\backslash X$. Using the notation above, we may write these rules as follows:

(9) a.
$$\frac{\begin{array}{c} [Y]^i \\ \vdots \\ X \end{array}}{X/Y}\,/\mathrm{I}^i$$
b.
$$\frac{\begin{array}{c} [Y]^i \\ \vdots \\ X \end{array}}{Y\backslash X}\,\backslash\mathrm{I}^i$$

Note however that this notation does not embody the conditions that have been stated, namely that in /I Y is the rightmost undischarged assumption in the proof of X, and in \I Y is the leftmost undischarged assumption in the proof of X. In addition, L carries the condition that in both /I and \I the sole assumption in a proof cannot be withdrawn, so that no types are assigned to the empty string.

In the introduction rules, the meaning of the result is given by lambda-abstraction over the meaning of the discharged assumption, which can be represented by a variable of the appropriate type. The relationship between lambda-abstraction and function application is given by the law of *β-equality* in (10),

where $\alpha[\beta|y]$ means 'α with β substituted for y'. (See Hindley and Seldin 1986 for a full exposition of the typed lambda-calculus.)

(10) $(\lambda y[\alpha])\beta = \alpha[\beta|y]$

Since exactly one assumption must be withdrawn, the resulting lambda-terms have the property that each binder binds exactly one variable occurrence; we refer to this as the 'single-bind' property (van Benthem 1983). The rules in (9) are analogous to the usual natural deduction rule of *conditionalization*, except that the latter allows withdrawal of any number of assumptions, in any position.

The /I and \I rules are commonly used in constructions that are assumed in other theories to involve 'empty categories', such as (11):

(11) (John is the man) who Mary likes.

We assume that the relative clause modifies the noun "man" and hence should receive the type N\N. The string "Mary likes" can be derived as of type S/NP, and so assignment of the type (N\N)/(S/NP) to the object relative pronoun "who" allows the analysis in (12) (cf. Ades and Steedman 1982):

(12)
$$\dfrac{\dfrac{\dfrac{\dfrac{}{(N\backslash N)/(S/NP)}\quad \dfrac{\dfrac{\text{Mary}\quad \dfrac{\dfrac{}{(NP\backslash S)/NP}\quad [NP]^1}{NP\backslash S}\backslash E}{S}/I^1}{S/NP}}{N\backslash N}/E}{NP}/E}{S}$$
(who: $(N\backslash N)/(S/NP)$; likes: $(NP\backslash S)/NP$)

The meaning of the string can be read off the proof by interpreting /I and \I as lambda-abstraction, giving the term in (13):

(13) who $(\lambda x[(\text{likes } x)\ \text{Mary}])$

Note that this mechanism is only powerful enough to allow constructions where the extraction site is clause-peripheral; for non-peripheral extraction (and multiple extraction) we appear to need an extended logic, as described later.

DERIVATIONAL EQUIVALENCE AND NORMAL FORMS

In the above system it is possible to give more than one proof for a single reading of a string. For example, compare the derivation of "Mary likes John" in (7), and the corresponding lambda-term in (8), with the derivation in (14) and the lambda-term in (15):

(14)
$$\dfrac{\dfrac{\dfrac{\text{Mary}\quad \dfrac{\dfrac{\text{likes}\quad \dfrac{}{(NP\backslash S)/NP}\quad [NP]^1}{NP\backslash S}\backslash E}{S}/I^1}{S/NP}\quad \dfrac{}{NP}}{S}/E$$
(likes: $(NP\backslash S)/NP$; John: NP)

(15) $(\lambda x[(\text{likes } x)\ \text{Mary}])\ \text{John}$

By the definition in (10), the terms in (8) and (15) are β-equal, and thus have the same meaning; the proofs in (7) and (14) are said to exhibit *derivational equivalence*. The relation of derivational equivalence clearly divides the set of proofs into equivalence classes. We shall define a notion of *normal form* for proofs (and their corresponding terms) in such a way that each equivalence class of proofs contains a unique normal form (cf. Hepple and Morrill 1989).

We first define the notions of *contraction* and *reduction*. A contraction schema $R \triangleright C$ consists of a particular pattern R within proofs or terms (the *redex*) and an equal and simpler pattern C (the *contractum*). A reduction consists of a series of contractions, each replacing an occurrence of a redex by its contractum. A normal form is then a proof or term on which no contractions are possible.

We define the following contraction schemas: *weak contraction* in (16) for proofs, and *β-contraction* in (17) for the corresponding lambda-terms.

(16) a.
$$\dfrac{\dfrac{\overset{[Y]^i}{\vdots}\ X}{X/Y}/I^i \quad \vdots\ Y}{X}/E \quad \triangleright \quad \vdots\ Y \ \vdots\ X$$

b.
$$\dfrac{\vdots\ Y \quad \dfrac{\overset{[Y]^i}{\vdots}\ X}{Y\backslash X}\backslash I^i}{X}\backslash E \quad \triangleright \quad \vdots\ Y \ \vdots\ X$$

(17) $(\lambda y[\alpha])\beta \triangleright \alpha[\beta|y]$

From (10) we see that β-contraction preserves meaning according to the standard functional interpretation of typed lambda-calculus. Therefore the corresponding weak contraction preserves the semantic functional interpretation of the proof; in addition it preserves the syntactic string interpretation since the redex and contractum contain the same leaves in the same order. For example, the proof in (14) weakly contracts to the proof in (7), and correspondingly the term in (15) β-contracts to the term in (8). The results of these contractions cannot be further contracted and so are the respective results of reduction to *weak normal form* and *β-normal form*.

Weak contraction in L strictly decreases the size of proofs (e.g. the number of symbols in a contractum is always less than that in a redex), and β-contraction in the single-bind lambda-calculus strictly decreases the size of terms. Thus there is *strong normalization* with respect to these reductions: every proof (term) reduces to a weak normal form (β-normal form) in a finite number of steps. This has as a corollary (*normalization*) that every proof (term) has a normal form, so that normal forms are *fully* representative: every proof (term) is equal to one in normal form. Since reductions preserve interpretations, an interpretation of a normal form will always be the

same as that of the original proof (term). Thus restricting the search to just such proofs addresses the problem of derivational equivalence, while preserving generality in that all interpretations are found.

Proofs in **L** and single-bind lambda-terms (like the more general cases of intuitionistic proofs and full lambda-terms) exhibit a property called the *Church-Rosser* property,[1] from which it follows that normal forms are unique.[2]

For formulations of **L** that are oriented to parsing, defining normal forms for proofs provides a basis for handling the so-called 'spurious ambiguity' problem, by providing for parsing methods which return all and only normal form proofs. See Kõnig (1989) and Hepple (1990).

STRUCTURAL MODALITIES

From a logical perspective, **L** can be seen as the weakest of a hierarchy of implicational sequent logics which differ in the amount of freedom allowed in the use of assumptions. The highest of these is (the implicational fragment of) the logistic calculus **LJ** introduced in Gentzen (1936). Gentzen formulated this calculus in terms of sequences of propositions, and then provided explicit *structural rules* to show the permitted ways to manipulate these sequences. The structural rules are *permutation*, which allows the order of the assumptions to be changed; *contraction*, which allows an assumption to be used more than once; and *weakening*, which allows an assumption to be ignored. For a discussion of the logics generated by dropping some or all of these structural rules see e.g. van Benthem (1987).

Although freely applying structural rules are clearly not appropriate in categorial grammars for linguistic description, commutable, iterable and optional elements do occur in natural language. This suggests that we should have a way to indicate that structural operations are permissible on specific types, while still forbidding their general application. To achieve this we propose to follow the precedent of the *exponential* operators of Girard's (1987) linear sequent logic, which lacks the rules of contraction and weakening, by suggesting a similar system of operators called *structural modalities*. Here we shall describe a system of *universal* modalities, which allow us to deal with the logic of commutable, iterable and optional extractions.[3]

For each universal modality we shall present an elimination rule, and one or more 'operational rules', which are essentially controlled versions of structural rules. (Introduction rules can also be defined, but we omit these here for brevity and because they are not required for the linguistic applications we discuss.) Note that these operators are strictly formal devices and not geared towards specific linguistic phenomena. Their use for the applications described, which are suggested purely for illustration, may lead to overgeneration in some cases.[4]

COMMUTATION

The type ΔX is assigned to an item of type X which may be freely permuted. Δ has the following inference rules:

(18)
$$\frac{\Delta X}{X}\Delta\mathrm{E} \qquad \frac{\Delta X \quad Y}{Y \quad \Delta X}\Delta\mathrm{Prm} \qquad \frac{Y \quad \Delta X}{\Delta X \quad Y}\mathrm{Prm}\Delta$$

From these rules we see that an occurrence of an item of type X in any position may be derived from an item of type ΔX.

We may use this operator in a treatment of relativization that will allow not only peripheral extraction as in (19a), but also non-peripheral extraction as in (19b):

(19) a. (Here is the paper) which Suzy read.

b. (Here is the paper) which Suzy read quickly.

We shall generate these examples by assuming that "which" licenses extraction from *any* position in the body of the relative clause. We may accomplish this by giving "which" the type (N\N)/(S/ΔNP) (cf. the extraction operator $\uparrow$ of Moortgat (1988)). This allows the derivations in (20a–b) (see Figure 1), which correspond to the lambda-terms in (21a–b) respectively:

(21) a. which (λx[(read x) Suzy])

b. which (λx[(quickly (read x)) Suzy])

ITERATION

The type $X^!$ is assigned to an item of type X which may be freely permuted and iterated. $^!$ has the following inference rules:

(22)
$$\frac{X^!}{X}{}^!\mathrm{E} \qquad \frac{X^! \quad Y}{Y \quad X^!}{}^!\mathrm{Prm} \qquad \frac{Y \quad X^!}{X^! \quad Y}\mathrm{Prm}^! \qquad \frac{X^!}{X^! \quad X^!}{}^!\mathrm{Con}$$

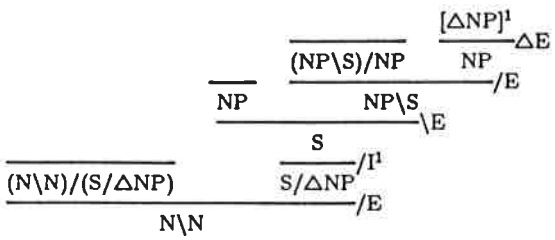
[1] This is the property that if a proof (term) M reduces to two proofs (terms) N_1, N_2, then there is a proof (term) to which both N_1 and N_2 reduce.

[2] The above remarks also extend to a second form of reduction, *strong reduction/η-reduction*, which we have not space to describe here. See Morrill *et al.* (1990).

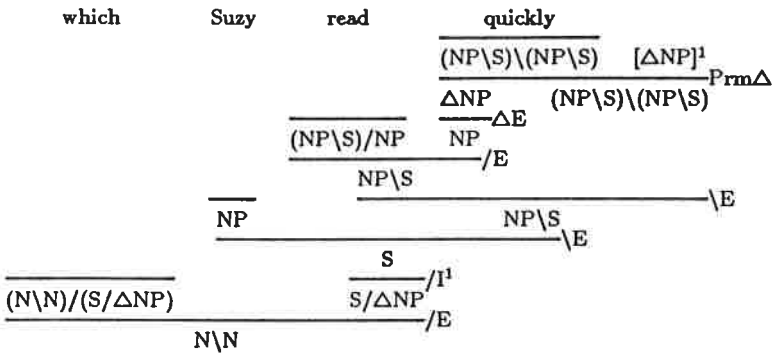
[3] The name is chosen because the elimination and introduction rules appropriate to each operator turn out to be those for the universal modality in the modal logic S4. See Dosen (1990).

[4] In Morrill *et al.* (1990) we give a system of modalities that differs from the present proposal in several respects. There are two unidirectional commutation modalities rather than the single bidirectional modality given here, and a *single* operational rule is associated with each of the universal modalities. We also suggest a (more tentative) system of *existential* modalities for dealing with elements that are themselves commutable, iterable or optional.

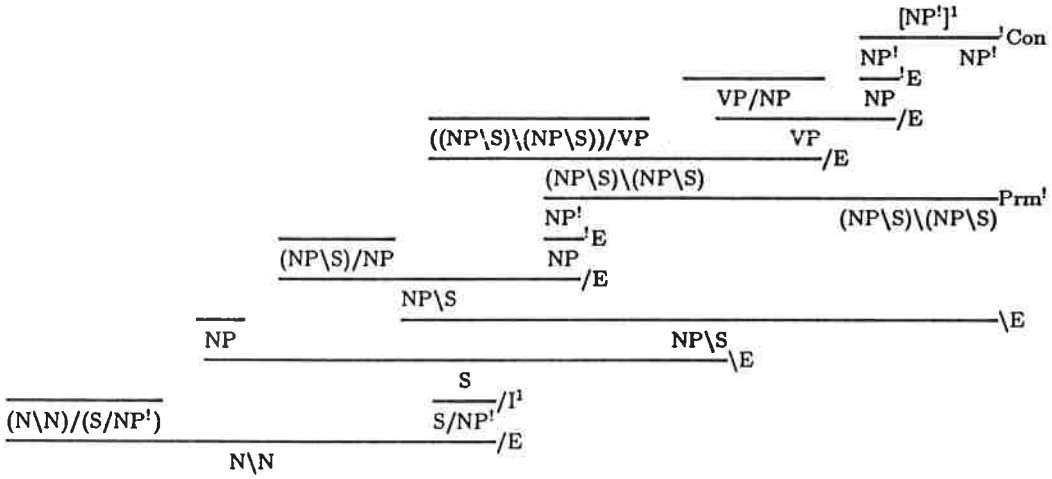
(20) a. which Suzy read



b. which Suzy read quickly



(24) which Suzy read without understanding



(28) too long for Suzy to concentrate

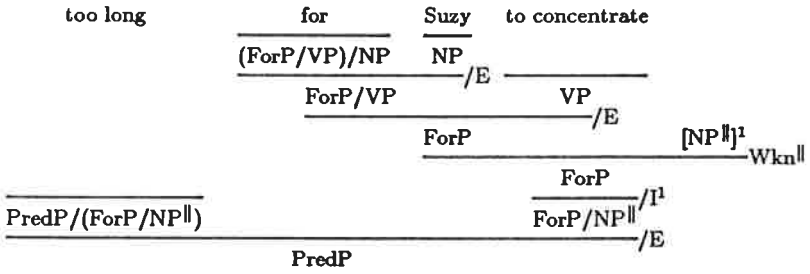


Figure 1. Derivations illustrating use of structural modalities

One or more occurrences of items of type X in any position may be derived from an item of type $X^{!}$.

We may use this modality in a treatment of multiple extraction. Consider the parasitic gap construction in (23):

(23) (Here is the paper) which Suzy read without understanding.

In order to generate both this example and the ones in (19), we shall now assume that "which" licenses extraction not just from any position in the body of a relative clause, but from *any number of* positions greater than or equal to one. We may do this by altering the type of "which" to $(N\backslash N)/(S/NP^{!})$. Since $^{!}$ has all the inference rules of Δ, the derivations in (20) will still go through with the new type. In addition, the $^{!}$Con inference rule allows the derivation of (23) given in (24) (see Figure 1), and the corresponding term in (25):

(25) which (λx[((without (understanding x)) (read x)) Suzy])

OPTIONALITY

The type $X^{\|}$ is assigned to an item of type X which may be freely permuted, iterated and omitted. $^{\|}$ has the following inference rules:

(26)
$$\frac{\vdots\ X^{\|}}{X}\,{}^{\|}\mathrm{E} \qquad \frac{\vdots\ X^{\|}\quad \vdots\ Y}{Y\quad X^{\|}}\,{}^{\|}\mathrm{Prm} \qquad \frac{\vdots\ Y\quad \vdots\ X^{\|}}{X^{\|}\quad Y}\,\mathrm{Prm}^{\|}$$

$$\frac{\vdots\ X^{\|}}{X^{\|}\quad X^{\|}}\,{}^{\|}\mathrm{Con} \qquad \frac{\vdots\ X^{\|}\quad \vdots\ Y}{Y}\,{}^{\|}\mathrm{Wkn} \qquad \frac{\vdots\ Y\quad \vdots\ X^{\|}}{Y}\,\mathrm{Wkn}^{\|}$$

Zero or more occurrences of items of type X in any position may be derived from an item of type $X^{\|}$.

We may use this modality in a treatment of optional extraction, as illustrated by (27):

(27) a. (The paper was) too long for Suzy to read.

b. (The paper was) too long for Suzy to read quickly.

c. (The paper was) too long for Suzy to read without understanding.

d. (The paper was) too long for Suzy to concentrate.

We shall assume for simplicity that "to"-infinitives are single lexical items of type VP, that "for-to" clauses have a special atomic type ForP (so that "for" has the type (ForP/VP)/NP), and that predicate phrases have a special atomic type PredP. Given these assignments, the type PredP/(ForP/NP$^{!}$) for "too long" would allow (27a–c), but not (27d). In order to generate all four examples, we shall assume that "too long" licenses extraction from any number of positions in the embedded clause greater than or equal to zero, and thus give it the type PredP/(ForP/NP$^{\|}$). Again, $^{\|}$ has all the inference rules of $^{!}$, generating (27a–c), and the Wkn$^{\|}$ rule allows (27d) to be derived as in (28) (see Figure 1), giving the term in (29):

(29) too-long (λx[for (to-concentrate Suzy)])

CONCLUSIONS

We have introduced a scheme of proof figures for Lambek categorial grammar in the style of natural deduction, and proposed structural modalities which we suggest are suitable for the capture of linguistic generalizations. It remains to extend the semantic treatment of the structural modalities, to refine the proof theory, and hence to develop more efficient parsing algorithms. For the present, we hope that the proposals made can be seen as gaining linguistic practicality in the categorial description of natural language, without losing mathematical elegance.

REFERENCES

Ades, A.E. and Steedman, M.J. (1982). On the order of words. *Linguistics and Philosophy 4*, 517–558.

van Benthem, J. (1983). *The semantics of variety in categorial grammar.* Report 83–29, Department of Mathematics, Simon Fraser University. Also in W. Buszkowski, W. Marciszewski and J. van Benthem (eds), *Categorial Grammar*, Volume 25, Linguistic & Literary Studies in Eastern Europe, John Benjamins, Amsterdam/Philadelphia, 57–84.

van Benthem, J. (1987). Categorial grammar and type theory. Prepublication Series 87–07, Institute for Language, Logic and Information, University of Amsterdam.

Dosen, K. (1990). Modal logic as metalogic. To appear in P. Petkov (ed.), *Proceedings of the "Kleene '90" Conference*, Springer-Verlag.

Gentzen, G. (1936). On the meanings of the logical constants. In Szabo (ed., 1969), *The Collected Papers of Gerhard Gentzen*, North Holland, Amsterdam.

Girard, J-Y. (1987). Linear logic. *Theoretical Computer Science 50*, 1–102.

Hepple, M. (1990). Normal form theorem proving for the Lambek calculus. In *Proceedings of COLING 1990.*

Hepple, M. and Morrill, G. (1989). Parsing and derivational equivalence. In *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, UMIST, Manchester.

Hindley, J.R. and Seldin, J.P. (1986). *Introdution to Combinators and Lambda-Calculus.* Cambridge University Press, Cambridge.

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

Kōnig, E. (1989). Parsing as natural deduction. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics.*

Lambek, J. (1958). The mathematics of sentence structure. *American Mathematical Monthly 65*, 154–170.

Moortgat, M. (1988). *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus.* Foris, Dordrecht.

Morrill, G., Leslie, N., Hepple, M. and Barry, G. (1990). Categorial deductions and structural operations. In G. Barry and G. Morrill (eds), *Edinburgh Working Papers in Cognitive Science, Volume 5: Studies in Categorial Grammar*, Centre for Cognitive Science, University of Edinburgh.

Prawitz, D. (1965). *Natural Deduction: a Proof Theoretical Study.* Almqvist and Wiksell, Uppsala.

Troelstra, A.S. (1969). *Principles of Intuitionism: Lecture Notes in Mathematics Vol. 95.* Springer-Verlag.