The categorial fine-structure of natural language

Johan van Benthem, Amsterdam & Stanford, March 2003

Abstract Categorial grammar analyzes linguistic syntax and semantics in terms of type theory and lambda calculus. A major attraction of this approach is its unifying power, as its basic function/argument structures occur across the foundations of mathematics, language and computation. This paper considers, in a light example-based manner, where this elegant logical paradigm stands when confronted with the wear and tear of reality. Starting from a brief history of the Lambek tradition since the 1980s, we discuss three main issues: (a) the fit of the lambda calculus engine to characteristic semantic structures in natural language, (b) the coexistence of the original type-theoretic and more recent modal interpretations of categorial logics, and (c) the place of categorial grammars in the complex total architecture of natural language, which involves - amongst others - mixtures of interpretation and inference.

1 From Montague Grammar to Categorial Grammar

Logic and linguistics have had lively connections from Antiquity right until today (GAMUT 1991). A recurrent theme in this history is the categorial structure of language and ontology, from Aristotle's grammatical categories to Russell's theory of types in the foundations of mathematics. Further bridges were thrown as logic and linguistics developed, with Lambek 1958 as a major advance in sophistication, many years ahead of its time. Large-scale logical analysis of natural language really took off around 1970, thanks to Montague's work (Montague 1975, Partee 1997). This system connects linguistic structure and logical semantics in the following architecture. The pivotal deep structure are syntax trees formed by grammatical construction rules. In one direction, these trees project to actual surface expressions by an obvious deletion map shedding all category symbols, brackets, indices, and other theoretical entities put there by the theory. In another, semantic direction, trees constructed in this explicit manner can be translated compositionally into logical formulas of a sufficiently rich language, for which Montague chose an intensional type logic *IL* (cf. Gallin 1975 for its mathematical theory). In particular, this sends syntactic categories to semantic types. These logical formulas have a standard compositional model-theoretic semantics, which then comes to natural language by composing the two homomorphisms. Alternatively, one can do model-theoretic interpretation directly on the syntax trees. In a schema, we have:

$$\begin{tikzcd}
\boxed{\text{Actual Expressions}} & \boxed{\text{Syntax Trees}} \arrow[l, "\textit{surface}"', "\textit{form}"] \arrow[r, "\textit{translation}"] & \boxed{\text{Logical Formulas}} \arrow[r, "\textit{semantic}", "\textit{denotation}"'] & \boxed{\text{Models Described}}
\end{tikzcd}$$

The expressive power of this apparatus is immense (Janssen 1983): it can recognize and interpret all *RE* languages, and hence it stands to natural language a bit like Turing machines stand to real computation. Thus, around 1980, a need was felt for significant *constraints* on what the Montague Machine can do. Mathematical linguistics already had natural grammar levels, from simple regular to complex *RE*. Could there be similar fine-structure parameters on the logico-semantic side? Indeed, the above architecture offers some footholds. Though its syntactic tree operations can be baroque, in practice, most of them amount to some form of concatenation with categorial function–argument composition. And likewise, the logical forms with their semantics suggest levels of expressive power inside full higher-order logic.

Then Categorial Grammar returned on stage. On the syntactic side, Buszkowski 1982 revived the study of Lambek grammars, pointing out their linguistic interest and mathematical depth in an influential series of papers (cf. Buszkowski 1997). This initiated a tradition running all the way to the architectures of explicit proof-theoretic control in Moortgat 1997, Morrill 1994, Gabbay, Kempson & Meyer Viol 2000, where different categorial inference rules and modal vocabularies are used to find the right combinatorial fit for linguistic phenomena. In the meantime, this stream has merged with linear logic, and substructural logics generally (Restall 2000). Simultaneously, around 1980, I was thinking about the minimal glue needed to compose linguistic meanings inside the baroque mechanics of Montague grammar. This led me to define a categorial calculus inspired by Geach 1972, associating single-occurrence lambda terms to derivations, and noting such nice combinatorial features as the invariance property for occurrence counts between premises and conclusions. I then learnt about Lambek 1958, and van Benthem 1983 shows how the Lambek derivations correspond one-one to special linear lambda terms. The now common term 'Lambek Calculus' seems due to me – though audiences always complained that it was indistinguishable from 'Lambda Calculus' at my supersonic rate of speaking. It took some more time to realize that this reinvented the Curry-Howard Isomorphism for a non-classical proof calculus – a move with precedents in relevant logic. But once all this was in place, questions of fine-structure could be asked, including the study of linguistic ambiguity: precise numbers of different readings for natural language expressions. The eventual result of my fateful encounter with Lambek was van Benthem 1991, which connects logic and grammar across a wide interface. In the present paper, I look at some unresolved issues and new themes.

2 Natural language in the light of type theory

Categorial calculi are close to mathematical type theories, and hence they suggest a proof-theoretic lambda-calculus based paradigm for natural language understanding. This match between mathematical logic and linguistics has proved natural and fruitful. But, as a general mechanism for the analysis of natural language, it also raises a number of open questions, all about the linguistic role of categorial proofs.

2.1 Categorial derivation and computing meanings

Categorial derivation in the Lambek Calculus has many formats, but at some level, all involve binary assertions of the form

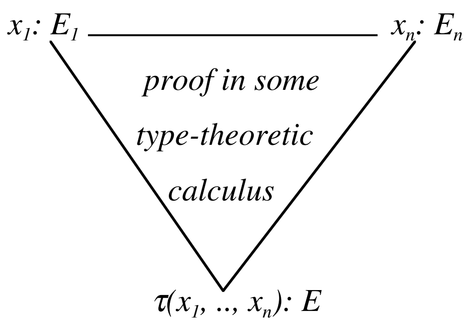
expression E has syntactic category C
term τ has semantic type a

Moreover, the two viewpoints work in tandem, so that a syntactic parse of a string of words with syntactic categories produces a matching description of a denotation, using correlated semantic types. This proof-theoretic parsing–as–deduction is computationally attractive, it works compositionally, and it gives the semantics for free. For instance, mixing syntactic and semantic types, the typed string feed $_{e \to (e \to t)}$ all $_{(e \to t) \to ((e \to t) \to t)}$ penguins $_{e \to t}$

amounts to a categorial derivation for the type $e \to t$ from these three premises, whose associated meaning recipe is given by the function composition term

$$\lambda x_e \bullet (ALL(PENGUINS))_{((e \to t) \to t)} \ (FEED_{e \to (e \to t)} (x_e))$$

In a simplified format, we can picture this analysis for a given string $E = \langle E_1, .., E_n \rangle$ using a proof tree with assertions σ:X saying that expression X has meaning σ:



We can also add further information to the assertions: syntactic, or even phonological. Thus, rich grammatical derivation becomes the well-established notion of formal proof. (Note 1.) Lambda calculus and type theory are thriving areas of mathematical logic (Barendregt 1992), whose theory can now be imported to natural language.

Moreover, these systems are not just engineering devices for combinatorial puzzles of syntax. They also provide a more general perspective on *parameters* for linguistic description: an explanatory desideratum in linguistic frameworks. The paradigm predicts that natural variations among grammars will occur by varying proof strength for the type theoretic calculi driving the above proof trees. This has been borne out most spectacularly by occurrence-based substructural calculi like that of Lambek 1958. Moreover, in addition to *deductive strength*, one can vary *expressive strength*, by changing the vocabulary of type-forming operations, yielding even more syntactic control (Moortgat 1997). Thus, the wide variety of human languages might come about by different parameter settings for deductive and expressive strength.

This view of what makes language tick is attractive, but it has not gone unchallenged. Does natural language really revolve around functional expressions hooking up to others via fixed slots? An alternative with a logical pedigree are *unification-based grammars*, which involve flat merging of equational constraints from diverse sources: syntactic, semantic, or even pragmatic. The equational logics driving this approach have a well-established tradition, too. (Note 2.) In the rest of this paper, we shall stick with the categorial approach, though many of our points apply more generally.

Confronting categorial logics with language structure raises special issues beyond engineering of the right categories and straight deduction – and so categorial grammar is not just a colony of type theory. We discuss a few of these issues (a few more are in the Notes), mainly concerning the semantic uses of the calculus. This is in line with Montague grammar, whose impact has been more on semantics than on syntax.

2.2 The proof steps: what is the glue?

The type-theoretic approach suggests that the only systematic *combinatorial glue* for composing natural language meanings out of their constituents is that provided by lambda terms, viz. function application and lambda abstraction. These are the logical counterparts of the proof steps of a categorial implication calculus, via the Curry-Howard isomorphism. Moreover, as can be seen in linguistic examples, this glue is restricted to typed terms whose lambdas bind *single occurrences* of variables. If natural language wants to bind more variable positions, it says so explicitly. E.g., to make one subject bind both arguments of a transitive verb "admire", it adds a *reflexivizer SELF*, as in the sentence 'Hyacinth admires herself'. *SELF* is an expression of type $(e\to(e\to t)) \to (e\to t)$ with a lexical lambda meaning

$$\lambda R_{(e\to(e\to t))}\bullet\ \lambda x_e\bullet\ R_{(e\to(e\to t))}\,(x_e)(x_e)$$

This is extra lambda glue that comes at a price, viz. explicit mention in the syntax.

But the issue of free combination is more complex than this. Some constructions in natural language suggest that we do get a bit of stronger logical vocabulary for free.

For instance, proper nouns in type e can serve as unary predicates in type $e \to t$ (as in German 'der Heinrich') which suggests free uses of *identity* to form singleton terms

$$\lambda y_e \bullet\ y_e = x_e$$

Likewise, some adjectives seem to have a natural meaning with a free Boolean *conjunction*. A "green bag" is something which is both green and a bag:

$$\lambda x_e \bullet\ GREEN_{e\to t}\ (x_e)\ \&\ BAG_{e\to t}\ (x_e)$$

A linguistic area which poses particular challenges of this sort are quantificational patterns in natural language. An evergreen like "Every boy loves a girl" can be analyzed in a straightforward categorial manner, by first composing "loves a girl" and then feeding this to the noun phrase "every boy". But many quantifier constructions defy this straightforward composition, in that they show emergent meanings. For instance, "Three managers owned ten Porsches" need not mean that there were three ten-Porsche owners: it can also mean that the three managers owned ten Porsches total. This so-called cumulative meaning comes for free: but what is its glue? Things get even more complex in a sentence like "The boys got one cookie each". This strongly suggests that there is a *one-to-one* map from boys to cookies, even though there is no overt identity in the sentence. The free glue needed to define all this again goes beyond simple lambda calculus, involving at least identity. Finally, free meaning composition of quantifiers gets even more complicated in the case of plural expressions and mass terms (van der Does 1992). Indeed, there is no definitive invariance analysis of quantifier constructions in natural language enumerating all these emergent meanings (van Benthem 1989, Keenan & Westerståhl 1997).

In more traditional logical terms, we do not yet understand the full *syncategorematic* repertoire of natural language, and hence, the logical glue to be put into its calculus. Single occurrence lambda binding is prominent in the glue, but we may also need some free uses of identity and Boolean conjunction. The latter items will return in what follows. There are still further unresolved aspects to the proof paradigm for natural language, such as the extent and purpose of the phenomenon of *ambiguity*, i.e., different derivations for the same string. But we omit these here. (Note 3.)

2.3 Proof data: semantic constraints on lexical items

Proofs consist of steps starting from certain initial givens. Proof steps suggested restrictions on combinatorial glue, but there may also be constraints on the initial materials to be glued together. Or linguistically, what meanings can occur for expressions has a lexical aspect just as much as a compositional one. Indeed, it has been found that denotations of basic expressions satisfy systematic *semantic constraints*. Indeed, the lambda calculus itself points toward one sort of these.

Invariance for transformations Pure lambda terms denote special type-theoretic objects which are invariant for permutations of objects in the universe of discourse. This is a general semantic constraint whose logical history goes back at least to Tarski (cf. van Benthem 1991). Note that permutations respect *identity* of objects, and hence the latter notion again emerges in the foundations of natural language semantics. Basic logical expressions like Booleans or quantifiers are permutation-invariant – but so are other syntactic expressions, like the above reflexivizer *SELF*. All this exemplifies a general perspective known from mathematics since the 19th century. Many linguistic categories show semantic invariances for suitable transformations of relevant objects. In general, these transformations may preserve much more structure than identity between objects. Examples are order-preserving transformations for tenses of temporal adverbs, or transformations respecting geometrical patterns for spatial prepositions. Van Benthem 2002 offers a general logical perspective on the invariance paradigm, its attractions and its foundational limitations. (Note 4.)

Boolean structure Another source of semantic constraints on linguistic expressions is the pervasive Boolean structure of natural language, which surfaced in our discussion of the glue. This time, consider implicit constraints on denotations for lexical items. E.g., determiner expressions D like "all", "two", "most", "enough" are *conservative*, in that their first argument restricts the range of the whole assertion:

$$D\ (A, B) \quad \text{iff} \quad D\ (A, B \cap A)$$

But Boolean structure is much more pervasive than this, as all domains for 'Boolean types' ending in a final truth value type *t* have a natural inclusion structure (Keenan & Faltz 1985). Many further semantic constraints that have been found involve Boolean structure. For instance, some linguistic expressions denote *Boolean homomorphisms* in their function type: an example is again the reflexivizer. (Note 5.) All this makes Boolean structure a serious candidate for inclusion in a base logic of natural language. Then we would need enriched typed lambda calculi with additional Boolean operators having their normal meaning. E.g., van Benthem 1991 extends Friedman's completeness theorem to such calculi – and still richer linear logics for linguistic syntax include additive conjunctions (Moortgat 1997). Still, it seems fair to say that, with a few exceptions (Dalrymple et al. 1995), such richer categorial systems have not yet addressed the full agenda of Montague-style semantics of natural language.

Through categorial derivation, all this information, lexical and compositional, gets combined in complex expressions. Many details of the total system are ill-understood. E.g., the vast function spaces of type theory have many exotic inhabitants. How can we zoom in more realistically on the denotations for actual linguistic expressions?

2.4 Parsing and inference: the case of monotonicity

Finally, Boolean structure also points at another aspect of natural language. Expressions are used to infer things, and logical calculi for interpreting natural language should also respect and explain the evidence about intuitively valid *inferences*. This broadens the range of linguistic phenomena which a logical calculus should deal with from syntax and semantic more narrowly construed – as has been realized very forcefully in the Montagovian tradition (cf. Partee & Hendriks 1997).

In particular, the natural relation of Boolean *inclusion* $\leq$ is like logical implication, which plays an important role throughout natural language semantics. For instance, a quantifier expression *"All (X, Y)"* allows the following valid inferences:

$$\frac{All\,(X, Y) \qquad Z \leq X}{All(Z, Y)} \qquad\qquad \frac{All(X, Y) \qquad Y \leq Z}{All(X, Z)}$$

We say that "All" is downward monotone with respect to inclusion in its *X*-argument, and upward monotone in its *Y*-argument. On the basis of such lexical observations, we must then explain more generally when expressions are monotone in which of their parts. For instance, the earlier phrase "feed all penguins" is upward monotone in the words 'feed' and 'all', but downward monotone in 'penguins'. A categorial calculus with suitable Boolean structure can account for this in a systematic manner (van Benthem 1986, 1991). This gives rise to open questions about monotonic lambda terms, which we will formulate in our discussion of 'natural logic' in Section 4.

This concludes our discussion of the fit between traditional categorial analysis and the intrinsic joints in the syntax and semantics of natural language. Our main claim is that, though the Lambek calculus does much of the job of meaning composition, there are quite a few systematic phenomena slightly beyond it, which seem of a logical nature all the same. Thus, the basic mechanism of free semantic composition might lie somewhat higher up in type theory – but at present, we do not know where.

We now broaden our discussion to other logical views on the categorial apparatus (Section 3), and after that, to a broader view of language as a whole (Section 4).

3 Emergence of the Modal Paradigm

Since the early 1990s, interest has also focused on another form of semantics for categorial calculi, namely models for which they are sound and complete. The atmosphere then changes from proofs and type theory to models for, in particular, *modal logics*. Roughly speaking, categorial operations may be viewed as binary modalities, and this view has been reinforced by the use of further modal operators over this base (cf. Moortgat 1997). In this section, we outline this model-theoretic semantics of categorial deduction, discuss some of its uses vis-à-vis natural language, and suggest some comparisons with the original type-theoretic approach.

3.1 A budget of modal models

There are many modal interpretations for categorial deduction. We note a few characteristic ones here – van Benthem 1991, 2003 present more complete overviews.

Language models Lambek's original syntactic calculus intuitively refers to models whose universe consists of all strings over some initial alphabet, while languages themselves are arbitrary sets of such strings. For instance, a categorial product $A{\bullet}B$ then denotes the obvious concatenation product of languages, while a categorial left implication $A{\rightarrow}B$ denotes a sort of left-looking functional string language:

$$L_{A\bullet B} \quad = \quad \{s^\frown t \mid s \in L_A,\ t \in L_B\}$$
$$L_{A\rightarrow B} \quad = \quad \{s \mid \textit{for all } t \in L_A\text{: } t^\frown s \in L_B\}$$

It took until Pentus 1993 before the associative Lambek Calculus LC was shown complete for this intended interpretation, in the following sense:

> A sequent $A_1, \ldots, A_k \Rightarrow B$ is derivable in LC if and only if for each interpretation L in a language model, the concatenation product of the premise languages is contained in the conclusion language.

Process models Other attractive models for categorial calculi have a more dynamic flavour, involving state transitions. For instance, categories may also be interpreted as *binary transition relations*, with categorial product becoming relational composition, and the directed arrows its left- and right-inverses. This leads to semantic clauses like:

$$R_{A\bullet B} \quad = \quad R_A\ ;\ R_B$$
$$R_{A\rightarrow B} \quad = \quad \{(s, t) \mid \textit{for all } (u, s) \in R_A\text{: } (u, t) \in R_B\}$$

Valid consequence now means that, in any such process model, the composition of the relations to the left of a sequent must be contained in the consequent relation. Completeness for process models was shown in Andréka & Mikulas 1993.

Vector models In the meantime, there are even vector models, in which category expressions denote sets of vectors, viewed as regions of a geometrical space.

Categorial operations are then identical with the Minkowski operations on images in mathematical morphology (Aiello & van Benthem 2002). Vector models generalize the numerical models introduced in van Benthem 1991, which were inspired by the occurrence count invariants of derivable categorial sequents. Each simple numerical vector model represents an invariant that can be used for pruning proof search trees.

Abstract ternary models The preceding models have a common generalization:

$$\boldsymbol{M} = (S, R, V)$$

for a minimal binary modal logic with a ternary accessibility relation *Rs, tu*, which can be read as "*s* is the concatenation of the strings *t, u*", "*s* is the composition of the transitions *t, u*", etc. The truth definition for an existential binary modality is this:

$$\boldsymbol{M}, s \models \langle\bullet\rangle\phi\psi \quad \text{iff} \quad \exists t, u\text{: } Rs, tu \;\&\; \boldsymbol{M}, t \models \phi \;\&\; \boldsymbol{M}, u \models \psi$$

To get the categorial implications, one needs three modalities describing *R*–compositions in different orders, a so-called versatile triple (Venema 1991):

$$\boldsymbol{M}, s \models A\bullet_1 B \quad \text{iff} \quad \exists tu\text{: } Rs, tu \;\&\; \boldsymbol{M}, t \models A \;\&\; \boldsymbol{M}, u \models B$$

$$\boldsymbol{M}, \mathrm{s} \models A\bullet_2 B \quad \text{iff} \quad \exists tu\text{: } Rt, us \;\&\; \boldsymbol{M}, u \models A \;\&\; \boldsymbol{M}, t \models B$$

$$\boldsymbol{M}, s \models A\bullet_3 B \quad \text{iff} \quad \exists tu\text{: } Rt, su \;\&\; \boldsymbol{M}, u \models A \;\&\; \boldsymbol{M}, t \models B$$

Ternary models validate a decidable minimal modal logic which embeds categorial calculi, as the three basic operations of categorial grammar translate as follows:

$$T(A\bullet B) = A\bullet_1 B$$

$$T(A\rightarrow B) = \neg(\neg A\bullet_2 B)$$

$$T(B\leftarrow A) = \neg(\neg A\bullet_3 B)$$

Further structure on the models is optional. E.g., the product relations will only become *associative* if we impose additional corresponding modal axioms. Without this assumption, the given translation validates precisely the non-associative Lambek Calculus – with it, we get the associative version *LC* (Kurtonina 1995).

Ternary models are very austere. Our final class of modal models is a common structure to the earlier examples which retains a bit more vivid intuitions.

Arrow models One can think of state transitions or categorial morphisms as abstract arrows allowing for composition. Modally, arrow models are of the form

$$\boldsymbol{M} = (A, C^3, R^2, I^1, V)$$

with A a set of abstract arrows carrying three structural predicates:

$C^3\, x,yz$	x is a composition of y and z
$R^2\, x,y$	y is a reversal of x
$I^1\, x$	x is an identity arrow

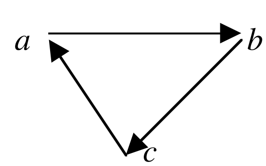
A suitable modal language has the following key clauses for two modalities:

$$\boldsymbol{M}, x \models \phi \bullet \psi \quad \text{iff} \quad \text{there are } y, z \text{ with } C\, x, yz \text{ and } \boldsymbol{M}, y \models \phi,\ \boldsymbol{M}, z \models \psi$$
$$\boldsymbol{M}, x \models \phi^{\smallsmile} \quad \text{iff} \quad \text{there exists } y \text{ with } R\, x, y \text{ and } \boldsymbol{M}, y \models \phi$$

There is again a minimal modal logic for such models in general (cf. van Benthem 1991, Venema 1996). On top of that, further axioms express constraints. In particular, assuming for convenience that reversal is a unary function r, two frame correspondences regulate the interaction of reversal and composition:

$$(\phi \bullet \psi)^{\smallsmile} \rightarrow \psi^{\smallsmile} \bullet \phi^{\smallsmile} \quad \text{iff} \quad \forall xyz\colon\ C\, x, yz \rightarrow C\, r(x), r(z)r(y)$$
$$\phi \bullet \neg(\phi^{\smallsmile} \bullet \psi) \rightarrow \neg\psi \quad \text{iff} \quad \forall xyz\colon\ C\, x, yz \rightarrow C\, z, r(y)x$$

Given this, there is no need for separate modal products any more, as we can view composition triangles like this from any arrow we please taking reversals:



The second reversal/composition law involves a sort of implication, reminiscent of the basic categorial application law $A \bullet A{\rightarrow}B \Rightarrow B$. And indeed, the above categorial to modal translation extends naturally to this setting:

Categorial $A{\rightarrow}B$ translates into arrow-logical $\neg(A^{\smile} \bullet \neg B)$

Kurtonina 1995 shows how categorial logics become arrow logics in this way, and sometimes vice versa. But note how categorial combination now involves arrow reversal. Thus, a dynamic arrow view of categorial grammar shifts intuitions.

This concludes our brief tour of modal models for categorial calculi. All examples given here generalize the original syntactic language models. But the approach can also analyze semantic models like the type hierarchies of Montague semantics. (Note 6.) As all these categorial models are removed from what modal logicians usually study, they also offer nice open questions for logic – whatever their linguistic merits. We now discuss some general features of the modal approach as such.

3.2 The modal worldview

The above models place categorial calculi in the setting of general modal logic. Thus, categorial languages may be viewed as fragments of modal languages whose formulas describe properties of strings, vectors, arrows, etc. Thus, existing modal techniques will apply. One of these is the systematic use of *translations* from a modal language into the first-order language over the relevant models. Here is how this would work out for categorial expressions over ternary models (Kurtonina 1995):

$A \bullet B$	$\exists yz\!: Rx, yz \;\&\; Ay \;\&\; Bz$
$A \rightarrow B$	$\forall yz\, ((Ry, zx \;\&\; Az) \rightarrow By)$
$A \leftarrow B$	$\forall yz\, ((Ry, xz \;\&\; Az) \rightarrow By)$

The translation is easily extended to further categorial operators, such as conjunction. Note the 'guarded' form of the first-order quantifiers matching the modal operators. This explains the modal semantic characteristic of invariance for bisimulation, and the decidability of the modal consequence problem (Andréka et al. 1998). Practically, the translation allows us to view issues in categorial models in the light of standard first-order model theory. Our next section has some examples involving model-theoretic preservation properties of categories and types.

More theoretically, the translation puts categorial languages in the following light. Basic modal logic is a decidable miniature version of first-order logic, and modal logic in general is about language design maintaining a balance between expressive power and complexity of satisfiability and other important tasks like model checking (Blackburn, de Rijke & Venema 2001). Even so, satisfiability in the minimal modal logic is *Pspace*-complete, and for the complete guarded fragment of first-order logic, it is even *Exptime*-complete. In this top-down perspective, categorial formalisms go one step further: they are tractable miniatures of modal languages. E.g., derivability in the non-associative Lambek Calculus goes down to time complexity *P*. (Notes 7, 8.) On the semantic side, categorial languages can be analyzed using invariance for modal-style bisimulations, with special twists reflecting the absence of Booleans (Kurtonina & de Rijke 1997). The resulting landscape of first-order fragments with varying deductive strength and vocabulary has not yet been charted systematically. Cf. Areces 2000 for the related case of hybrid logics and description logics, and Kerdiles 2001 for other non-Boolean fragments of first-order logic.

Finally, like the type-theoretic paradigm, our modal approach has competitors in linguistic analysis. For instance, the *modal grammar* of Blackburn 1995 uses various modal languages to describe syntax trees as the primary structures of linguistics. This is in the spirit of more radical model-theoretic proposals for redefining the business of linguistics as providing logical theories of linguistic structures (cf. Rogers 1996).

3.3 An illustration: learning

We now consider a typical new research topic from the 1990s, partly for its intrinsic interest, but mainly to show the above model-theoretic perspective at work.

Learning Categorial parsing presupposes that lexical expressions already have categories or types. The only issue is how to interpret a given combination of these, by finding the right categorial derivation. But initially, such category assignments have to be learnt before a language user reaches a steady state of competence.

In recent years, linguists have become increasingly interested in issues of learning. It is one thing to describe a functioning syntactic calculus, but another how to explain why it might be learnable. Without some such explanation, a grammatical paradigm lacks credibility. These issues have also occurred intermittently in categorial grammar. Kanazawa 1994 shows that standard categorial languages are not learnable in the sense of Gold, while suitable modifications are. Vervoort 2000 presents practical learning algorithms with statistical twists due to Adriaans 1992 working with some success on text corpora such as the Bible book of Genesis. Indeed, similar learning concerns should make sense for logical calculi of ordinary inference! We make a few semantic observations on parsing versus learning in a categorial setting.

Learning categories Categorial learning algorithms work on growing finite input sets, computing preliminary types for basic symbols at each stage. What guarantee is there that this process will stabilize, in the sense that we can determine whether a type holds or not at some finite stage, and stick to that? None, in general, because of the following semantic analysis in a modal language of categories.

Consider some categorial language model, representing the set of assertions known to us so far. We are given that some expressions have the basic type t, while for all others, we can compute whether they have any complex type a formed from t using products and implications by the earlier truth definition in language models. Suppose that in this way an object x is seen to have type $t{\to}t$. Now we add objects to the model, corresponding to new assertions coming in. Then the type of that object may change. Here is an illustration, involving the *first-order* type $t{\to}t$:

After this, no further change occurs: adding new objects will not change the truth value of $\neg\, t{\to}t$. But more complex cases can show alternating behaviour. Consider the *second-order* type $(t{\to}t) \to t$. This may be false for x in a model because there is a cheap isolated $t{\to}t$ object y to its left, without the concatenation of y, x being t. But now add an object z to y's left blocking its property $t \to t$ like above. Then $(t \to t) \to t$ becomes true again for x. But adding another isolated $t \to t$ object to its left makes it false again, etc. In terms of our learning procedure, a second-order type of the form $(t \to t) \to t$ may flip-flop all the time as we learn new facts about our language.

Category assignment and persistence These phenomena have a simple explanation in first-order logic. First note that our ternary first-order translation takes all non-nested first-order types – or rather, their corresponding modal formulas – into *universal* first-order formulas of the form 'prefix of universal quantifiers – quantifier-free part'. E.g., $t \to t$ becomes

$$\forall yz \ ((Ry, zx \ \& \ Tz) \ \to Ty)$$

These formulas stay false under model extension once they are falsified somewhere. Indeed, modulo equivalence, the first-order *existential* formulas are precisely those preserved under model extensions by the Los–Tarski Theorem. But second-order types are of a *universal-existential* form. E.g., $(t \to t) \to t$ translates into

$$\forall uv \ (\ (Ru, vx \ \ \& \ \ \forall yz \ \ ((Ry, zv \ \& \ Tz) \ \to Ty)) \ \to \ Tu)$$

(do not confuse the categorial and first-order arrows here!) which is equivalent to

$$\forall uv \ \exists yz \ (\ (Ru, vx \ \ \& \ \ ((Ry, zv \ \& \ Tz) \ \to Ty)) \ \to \ Tu)$$

The latter formulas, too, have a model-theoretic characterization. Consider a family of models which is *convergent*: any two models in the family are submodels of another in it. These families represent coherent investigations, say samples of some language. The natural *union* of such a family represents the total information obtained. Now, a first-order formula is preserved under unions of convergent families iff it is definable by a universal-existential formula (cf. Doets 1996). Thus, first-order and second-order types have some natural semantic behaviour. (Note 9.) These seem the most complex types encountered in natural language (cf. van Benthem 1991). These observations seem the model-theoretic side of the learning situation for categorial grammars.

Other issues in categorial practice can also be analyzed in this model-theoretic manner. Another interesting example would be categorial *parsing* and its systematic combination of syntactic and semantic structure. (Note 10.)

3.4 Connecting type theory and modal logic

Perhaps the most obvious methodological question in the area is how type-theoretic and modal perspectives are related. There have been various attempts at answering this issue, which emerges in many fields besides categorial grammar, such as intuitionistic logic: cf. Alechina et al. 2001, de Paiva 2002. In the categorial setting, derivations involve lambda terms τ which denote objects in semantic type domains. These objects are a much richer sort of denotation than that provided by the above models of strings or arrows. On the other hand, both realms have a parallel binary semantic format stating that objects have abstract unary properties:

object τ lives in type domain A
object s satisfies modal formula A.

Van Benthem 1998 suggests that the two meet at the level of ternary composition, reading Rs, tu as "value s is the result of applying function t to argument u". But this hides some important differences, such as the fact that validity does not have the inclusion character in the above modal models. For instance,

validity of a sequent $A_1, .., A_k \Rightarrow B$ says that there exists some typed lambda term τ of type B with only the free type variables $x_{A1}, .., x_{Ak}$.

Thus, the objects exemplifying the premises must be *transformed* into one exemplifying the conclusion. In this light, modal models are projections of type hierarchies, but the connection remains to be spelled out. On the other hand, arrow models suggest categories, which are the natural generalization of type hierarchies (cf. Lambek & Scott 1986). In this way, it might be possible to match up the category-theoretic analysis of categorial logics with modal perspectives. (Note 11.)

4 Inferential Architecture of Natural Language

Our final topic is the position of categorial grammar in the larger picture of logic and language. As usual, we discuss only a few features and unresolved questions.

Inference at various levels Categorial grammar views parsing as a form of inference. Now parsing is a fast, largely unconscious process, prior to the many conscious linguistic processes we engage in: planning an assertion, asking a question, or: drawing a conclusion. Natural language is full of inferential mechanisms for such diverse purposes. As is sometimes said, it has a *natural logic*. Some of these mechanisms play a role at the level of unconscious sentence interpretation. E.g., Kamp & Reyle 1993 show how analyzing and generating correct plural expressions involves semantic inferences about whether terms denote individuals or groups. (When Anjuli says "Farewell, my Lord and Light.", we know from the setting that a singular verb form must follow.) Other sorts of inference, the ones in our logic textbooks, drive conscious planned reasoning – and yet others lie in between. Thus, natural logic is an intriguing cognitive phenomenon straddling the fence between conscious and unconscious processes in language use, spread out over various mechanisms, with an architecture quite unlike standard logical systems.

Natural logic There is no established architecture for natural logic. Van Benthem 1987 proposes a number of levels. The simplest level are *monotonicity inferences* like

Most days are rainy, all rainy days are cloudy, therefore: most days are cloudy	$\dfrac{Q\,AB \quad B \subseteq B'}{Q\,AB'}$

These replace predicate occurrences by ones with a larger, or smaller extension. Inferences enlarging or shrinking predicates abound in linguistic interpretation, but they also occur in simple computational manipulation of data bases.

Another general mechanism is *domain restriction* for certain argument positions. This is witnessed by the earlier conservativity law for linguistic determiners:

$$Q\,(A, B) \quad \text{iff} \quad Q\,(A, B \cap A)$$

Here, the second argument is restricted by the first. More generally, natural language never seems to employ the unbounded Fregean quantifiers: every variable carries its own special sortal conditions. Domain restriction constrains variables, much as in the computer science literature on constraint satisfaction techniques.

Further natural logic mechanisms include more global interpretative processes such as the setting of *temporal perspective* in a narrative. For instance, ter Meulen 1995 shows how tenses and aspectual expressions help create an ordered sequence of events, where positioning of events is not just determined by syntax, but also by compatibility relations between predicates. E.g., "Hortensia entered and left" must be sequential in time, whereas "Hortensia entered and brought a novel" seems simultaneous. Moreover, this compatibility may depend on inferential long-distance effects: if Mary has died, then the information that she is dead remains available when interpreting syntax describing later events. Van Benthem & ter Meulen 1999 show how these inferences do not involve a full-fledged temporal logic as used in systems of conscious temporal planning or reasoning, but rather a tractable Horn-clause fragment of the latter. This observation also applies to our main topic.

The role of categorial calculi Categorial derivation has several roles to play in natural logic. First, resource-sensitive Lambek calculi are tractable unconscious *base mechanisms* of interpretation, making sense of linguistic expressions. But next, the very fact that there is no unique best calculus, but only a landscape of options, turns out to be a virtue. The exciting feature of the substructural landscape of categorial systems, all the way up to the contraction rule, is this. More classical calculi correspond to more time-consuming conscious inferential processes. After all, we encounter them when analyzing ordinary constructive inference. (Note 12.) Thus, the same system can be *parametrized* for different cognitive functions.

Finally, categorial systems also serve another useful function, because of their glue. They *spread information* from other sources through expressions. One nice example is monotonicity. Categorial construction lifts simple inferences to more complex settings, providing lambda terms showing, e.g., how an expression with a quantifier occurring under a preposition like "opens with a knife"

eventually gets positive monotonicity marking on all four of its constituent words. Likewise, categorial derivation systematically spreads information about variable restrictions (van Benthem 1991). This is how we get from conservativity of basic determiners to a valid equivalence in transitive sentences like

$$(Q_1\,A)\; R\; (Q_2\,B) \qquad \text{iff} \qquad (Q_1\,A)\; R \cap AxB\;\; (Q_2\,B)$$

in which A restricts the first argument of the binary verb, and B the second. An example would be the evergreen "Every man loves a woman", where the first argument of "loves" is restricted to men, and the second to women.

The case of monotonicity Interfacing categorial derivation with other inferential processes also raises some systematic questions. The following case has been studied extensively in van Benthem 1986, 1991. Monotonicity is such a simple inference because it takes a free ride on categorial derivation by the natural inductive definition of *positive occurrence in lambda terms*. One marks positions in expressions as positive or negative as they are parsed, virtually without cost, using an interplay of two sorts of information. One is the given monotonicity behaviour of lexical expressions: e.g., the left-down, right-up monotonic behaviour of a quantifier "all". The other is the monotonicity transfer due to the compositional glue. This makes all function heads and lambda bodies positive. By contrast, argument positions become 'opaque', blocking inference from them – unless they are opened up by functional terms having a monotonicity marking in their argument position.

Example "All boys fight" is upward monotone in "All" for general reasons: this is true for every sentence of the form "Q boys fight". It is also upward monotone in "fight", but this is due to lexical information about "All". E.g., no monotonicity marking, up or down, would occur in "Exactly ten boys fight". Another illustration of this interplay is the earlier sentence "feed all penguins" in Section 2:

$$\lambda x_e \bullet (ALL(PENGUINS))_{((e \to t) \to t)}\; (FEED(x))$$

Its categorial derivation makes the term *ALL* positive automatically, while making *FEED* upward monotonic and *PENGUINS* downward monotonic through adding lexical information about *ALL* at just the right places.

Thus, inferential sensitivity for monotonic replacement is part of our understanding an expression just as much as understanding its syntactic structure. Observations like this also raise technical logical questions. Recall the *Lyndon preservation theorem* in first-order logic. First, there is soundness: any formula $\phi(P)$ with only positive syntactic occurrences of the predicate P defines an operation which is semantically monotone in P. Lyndon also proved a completeness result saying that all semantic monotonicity comes about by positive occurrence (modulo logical equivalence). The same issue arises here. Let us say that a lambda term $\tau(x)$ is semantically *monotone* in the argument x – with some abuse of notation - whenever the following implication holds in every semantic model interpreting τ, with '≤' standing for Boolean inclusion:

if $x \leq y$, then $\tau(x) \leq \tau(y)$

The question whether an analogue of the Lyndon preservation theorem for first-order logic holds for the typed lambda calculus, is still open. Van Benthem 1991 does prove that semantic monotonicity implies positive syntactic definability for the *linear fragment* of the full lambda calculus which corresponds to derivations in the Lambek Calculus *LC*. But that proof proceeds by brute force, and does not generalize.

Sanchez Valencia 1991, 2001 show how such simple monotonicity calculi underlie most of traditional pre-Fregean syllogistics, and also find surprising parallels between the types of inference in C.S. Peirce's existential graph calculus of first-order reasoning. Moreover, these books provide a range of linguistic applications.

Combining systems Natural logic splits inference into a family of special purpose systems. It seems implausible that one simple logical idea, no matter how elegant mathematically, will unify all of these. But then, unity may also come about in different ways. A typical them in modern (modal) logic is the issue of *combining systems*. Instead of designing vast 'super-logics', one has a network of simple inferential systems, with suitable links allowing them to pass information as needed. (Cf. Gabbay 1996.) Thus, the logical unity is in the system of combination, rather than in one fundamental base calculus. The same architecture seems attractive for understanding natural language. Not much is known systematically about logical theory combinations and their properties – a topic which has received more attention in the philosophy of science than in logic proper. (Note 13.)

5 Conclusion

The main points of this paper can be summarized as follows. As briefly related in Section 1, categorial grammar and categorial logic provide elegant fine-structure to the logical study of natural language. Section 2 showed that there are still issues though about the fit between Lambek calculi and the free non-lexicalized composition mechanisms of natural language. Section 3 demonstrated that the framework satisfies one criterion for successful theories in science. These should have new interpretations completely unenvisaged when first proposed. The modal semantics amply serves this purpose, even though its precise relation to the categorial proof-theoretic perspective remains to be understood. Finally, Section 4 discussed the place of categorial logic in a larger view of interpretation and inference natural language, which is of course what one would like to understand eventually.

Here is one final thought. Language is an empirical cognitive phenomenon, and one with a lot of perhaps accidental history in its cultural genes. What can be the role of a simple uniform logical or mathematical paradigm in this setting? Returning to our link with Montague grammar, Montague's famous thesis stated that

there is no difference of principle between natural and formal languages

(cf. van Benthem 2002). What is being claimed here? Probably, this was meant as a dogma of methodology, not in any naturalistic sense. Thus, categorial logic might then just be a recommended mathematical working format for describing languages, say, like using differential equations in suitable applied domains. There need not be any more radical naturalistic claim that natural language is really like basic categorial mechanisms. But sometimes, much more is being claimed. An example is Macnamara

& Reyes 1994, a vigorous defence of type structure and category theory as the key to cognitive science, coming from Jim Lambek's own home base. This reflects an emerging broader discussion today about possible naturalistic uses of logical systems, calling into question Frege's celebrated anti-psychologism. But even in its underspecified reading, the Thesis has been immensely useful as a bridge between natural and formal languages. Indeed, by now, one would also include *programming languages* in the equation (cf. Janssen 1983). All this has led to a plethora of parallels between linguistics, philosophy, and computer science. Moreover, the computer science connection suggests an interesting broader use of categorial logic. Computer science often *creates* its own virtual reality to match its theories. Even where it does not match natural language perfectly, categorial grammar might suggest design of perspicuous and useful new languages, perhaps hybrids of formal and natural ones.

In all, I believe that the love match brokered by Lambek between categorial grammar and natural language still has some romance to it, even though there lots of interesting twists, misunderstandings and subplots to go before the happy ending.

6 References

P. Adriaans, 1992, *Language Learning from a Categorial Perspective*, dissertation, Institute for Logic, Language and Computation, University of Amsterdam.

M. Aiello & J. van Benthem, 2002, 'A Modal Walk through Space', in P. Balbiani, ed., issue of *Journal of Applied Non-Classical Logics* on spatial reasoning.

N. Alechina, M. Mendler, V. de Paiva & E. Ritter, 2001, 'Categorical and Kripke Semantics for Constructive Modal Logics', in L. Fribourg, ed., *Proceedings of the 15th International Workshop on Computer Science Logic*, Springer, Berlin.

H. Andréka & S. Mikulas, 1993, 'The Completeness of the Lambek Calculus with Respect to Relational Semantics', *Journal of Logic, Language and Information* 3, 1-37.

H. Andréka, J. van Benthem & I. Németi, 1998, 'Modal Logics and Bounded Fragments of Predicate Logic', *Journal of Philosophical Logic* 27:3, 217–274.

C. Areces, 2000, *Logic Engineering*, dissertation DS 2000–05, Institute for Logic, Language and Computation, University of Amsterdam.

H. Barendregt, 1992, 'Lambda Calculi with Types', in S. Abramsky, D. Gabbay & T. Maibaum, eds., *Handbook of Logic in Computer Science*, Clarendon Press, Oxford, 117-309.

J. Barwise & J. Seligman, 1995, *Information Flow*, Cambridge University Press, Cambridge.

J. van Benthem, 1983, 'The Semantics of Variety in Categorial Grammar', Report 83-26, Department of Mathematics, Simon Fraser University, Burnaby (B.C.).

J. van Benthem, 1986, *Essays in Logical Semantics*, Reidel, Dordrecht.

J. van Benthem, 1987, 'Meaning: Interpretation and Inference', *Synthese* 73, 451-470.

J. van Benthem, 1989, 'Polyadic Quantifiers', *Linguistics and Philosophy* 12:4, 437-464.

J. van Benthem, 1991, *Language in Action: categories, lambdas and dynamic logic*, Elsevier Science Publishers, Amsterdam. Reprint with Addenda, 1995, The MIT Press, Cambridge (Mass.).

J. van Benthem, 1995, ' Quantifiers in the World of Types'. In J. van der Does & J. van Eijck, eds., *Quantifiers, Logic and Language*, CSLI Publications, Stanford, 47 – 62.

J. van Benthem, 1998, 'Proofs, Labels, and Dynamics in Natural Language'. In U. Reyle & H-J Ohlbach, eds., *Festschrift for Dov Gabbay*, 31–41, Kluwer Academic Publishers, Dordrecht.

J. van Benthem, 2000, 'Linguistic Grammar as Dynamic Logic'. In V. M. Abrusci & C. Casadio, eds., *Dynamic Perspectives in Logic and Linguistics*, Bulzoni, Roma, 7–17.

J. van Benthem, 2002, 'Invariance and Definability: two faces of logical constants', in W. Sieg, R. Sommer, & C. Talcott, eds., *Reflections on the Foundations of Mathematics. Essays in Honor of Sol Feferman*, ASL Lecture Notes in Logic 15, 426–446.

J. van Benthem, 2002, 'Mathematical Logic and Natural Language'. In B. Löwe, W. Malzkorn, & T. Rasch, eds., *Foundations of the Formal Sciences II: Applications of Mathematical logic in Philosophy and Linguistics*, Kluwer, Dordrecht, 25–38.

J. van Benthem, 2003, 'Categorial Grammar at a Cross-Roads'. To appear in R. Oehrle & G-J Kruyff, eds., *Proceedings Workshop on Type Logic at ESSLLI Utrecht 2000*.

J. van Benthem, W. Buszkowski, & W. Marciszewski, eds., 1988, *Categorial Grammar*, John Benjamins, Amsterdam and Philadelphia.

J. van Benthem & A. ter Meulen, 1999, 'Dynamics of Interpretation versus Dynamics of Evaluation', *Proceedings 12th Amsterdam Colloquium*, ILLC Amsterdam.

P. Blackburn, 1995, 'Static and Dynamic Aspects of Syntactic Structure', *Journal of Logic, Language and Information* 4. 1–4.

P. Blackburn, M. de Rijke & Y. Venema, 2001, *Modal Logic*, Cambridge University Press, Cambridge.

W. Buszkowski, 1982, *Lambek's Categorial Grammars*, dissertation, Mathematical Institute, Adam Mickiewicz University, Poznan.

W. Buszkowski, 1997, 'Mathematical Linguistics and Proof Theory'. In J. van Benthem & A. ter Meulen, eds., *Handbook of Logic and Language*, Elsevier Science Publishers, Amsterdam, 638–738.

M. Dalrymple, J. Lamping, F. Pereira & V. Saraswat, 1995, 'Linear Logic for Meaning Assembly', in G. Morrill & R. Oehrle, eds., *Formal Grammar*, Proceedings of the 6th European Summer School of Logic, Language and Information, Barcelona 1995, 75–93.

J. van der Does, 1992, *Applied Quantifier Logics: Collectives and Naked Infinitives*, dissertation, Institute for Logic, Language and Computation, University of Amsterdam.

K. Doets, 1996, *Basic Model Theory*, CSLI Publications, Stanford.

J–E Fenstad, P-K Halvorsen, T. Langholm & J. van Benthem, 1987, *Situations, Language and Logic*, Reidel, Dordrecht.

D. Gabbay, 1996, *Labelled Deductive Systems*, Oxford University Press, Oxford.

D. Gabbay, R. Kempson & W. Meyer Viol, 2000, *Dynamic Syntax*, *the flow of language understanding*, Blackwell, Oxford.

D. Gabbay & V. Shehtman, 1998, 'Products of Modal Logics, Part 1', *Logic Journal of the IGPL* 6, 73-146.

D. Gallin, 1975, *Systems of Intensional and Higher-Order Modal Logic*, North-Holland, Amsterdam.

GAMUT, 1991, *Logic, Language and Meaning*, Chicago University Press, Chicago.

P. Geach, 1972, 'A Program for Syntax'. In D. Davidson & G. Harman, eds., *Semantics of Natural Language*, Reidel, Dordrecht 483-497.

Th. Janssen, 1983, *Foundations and Applications of Montague Grammar*, dissertation, Mathematical institute, University of Amsterdam.

H. Kamp & U. Reyle, 1993, *From Discourse to Logic*, Kluwer, Dordrecht.

M. Kanazawa, 1994, *Learnable Classes of Categorial Grammars*, dissertation, Department of Linguistics, Stanford University. ILLC series 1994-08.

E. Keenan & L. Faltz, 1985, *Boolean Semantics for Natural Language*, Reidel, Dordrecht.

E. Keenan & D. Westerståhl, 1997, 'Generalized Quantifiers'. In J. van Benthem & A. ter Meulen, eds., *Handbook of Logic and Language*, Elsevier Science Publishers, Amsterdam, 839–893.

G. Kerdiles, 2001, *Saying it with Pictures, a logical landscape of conceptual graphs*, dissertation DS-2001-09, Institute for Logic, Language and Computation, University of Amsterdam.

N. Kurtonina, 1995, *Frames and Labels*, dissertation, OTS, University of Utrecht & ILLC, University of Amsterdam.

N. Kurtonina & M. de Rijke, 1997, 'Bisimulations for Temporal Logic', *Journal of Logic, Language and Information* 6, 403-425.

M. van Lambalgen, 1995, 'Natural Deduction for Generalized Quantifiers'. In J. van der Does & J. van Eijck, eds., *Quantifiers, Logic and Language*, CSLI Lecture Notes, vol. 54, Stanford University, 225–236.

J. Lambek, 1958, 'The Mathematics of Sentence Structure', *American Mathematical Monthly* 65, 154-170.

J. Lambek & P. Scott, 1986, *Introduction to Higher Order Categorical Logic*, Cambridge University Press, Cambridge.

J. Macnamara & G. Reyes , eds., 1994, *The Logical Foundations of Cognition*, Oxford University Press, New York.

A. ter Meulen, 1995, *Representing Time in Natural Language*, MIT Press, Cambridge (Mass.).

R. Montague, 1975, *Formal Philosophy*, Cornell University Press.

M. Moortgat, 1997, 'Categorial Type Logics'. In J. van Benthem & A. ter Meulen, eds., *Handbook of Logic and Language*, Elsevier Science Publishers, Amsterdam, 93–177.

G. Morrill, 1994, *Type Logical Grammar*, Kluwer, Dordrecht.

V. de Paiva, 2002, 'Lineales: Algebras and Categories in the Semantics of Linear Logic', in D. Barker-Plummer, D. Beaver, J. van Benthem & P. Scotto di Luzio, eds., *Words, Proofs and Diagrams*, CSLI Publications, Stanford, 123-142.

P. Parikh, 2001, *The Use of Language*, CSLI Publications, Stanford.

B. Partee (with H. Hendriks), 1997, 'Montague Grammar'. In J. van Benthem & A. ter Meulen, eds., *Handbook of Logic and Language*, Elsevier Science Publishers, Amsterdam, 5–91.

M. Pentus, 1993, 'Lambek Grammars are Context-Free, *Proceedings 8th Annual Symposium on Logic in Computer Science*, 429–433.

G. Restall, 2000, *An Introduction to Substructural Logics*, Routledge, London.

J. Rogers, 1996, *A Descriptive Approach to Language-Theoretic Complexity*, CSLI Publications, Stanford.

W. Rounds, 1997, 'Feature Logics', In J. van Benthem & A. ter Meulen, eds., *Handbook of Logic and Language*, Elsevier Science Publishers, Amsterdam, 475–533.

V. Sanchez Valencia, 1991, *Studies on Natural Logic and Categorial Grammar*, dissertation, Institute for Logic, Language and Computation, University of Amsterdam.

V. Sanchez Valencia, 2001, *Natural Logic*, to appear with MIT Press, Cambridge (Mass.).

S. Shieber, 1986, *An Introduction to Unification-Based Approaches to Grammar*, CSLI Publications, Chicago University Press, Chicago.

E. Spaan, 2000, 'Poor Man's Modal Logic', in J. Gerbrandy, M. Marx, M. de Rijke & Y. Venema, eds., *JFAK, essays dedicated to Johan van Benthem on the occasion of his 50th birthday*, ILLC Amsterdam.

H-J Tiede, 2000, *Proofs and Trees - Deductive Systems and Grammars*, dissertation, department of cognitive science, Indiana University, Bloomington.

Y. Venema, 1991, *Many-Dimensional Modal Logic*, dissertation, Institute for Logic, Language and Computation, University of Amsterdam.

Y. Venema, 1996, 'A Crash Course in Arrow Logic', in M. Marx, M. Masuch & L. Pólos, eds., *Arrow Logic and Multimodal Logic,* Studies in Logic, Language and Information, CSLI Publications, Stanford, 3–34.

M. Vervoort, 2000, *Games, Walks, and Grammars*, dissertation DS 2000–03, Institute for Logic, Language and Computation, University of Amsterdam.

7 Notes

1 The equation of standard proof and linguistic meaning in our account does have a presupposition. Is the usual notion of equivalence of proofs or lambda terms really the right level of identity for assertions in natural language? This is not obvious, since there could be independent linguistic intuitions here, and perhaps even different notions of equivalence. There is little work on categorial proof equivalence and this linguistic issue of 'strong recognizing power', but cf. Tiede 2000.

2 Shieber 1986, Fenstad et al. 1987, and other authors propose *merging* of equational constraints as the basic linguistic mechanism. In this process, *unification* takes place of the relevant terms – and information flows as variables become more specified through successive matching. The resulting grammars, too, involve a computational framework with a wide spectrum of uses, viz. equational logic and constraint satisfaction. For some logical theory behind this paradigm, cf. Rounds 1977. Categorial and unification ideas co-exist in categorial unification grammars. These allow for function applications plus variable specification, as in the typical combination step from $(x\to t)\to y$ and $e\to y$ to t by unifying $x\to t$ and $e\to y$. The

underlying logics have both function application and unification in proofs. They are fragments of higher-order type theories with variable types (van Benthem 1991).

3 Different derivations of one type from a string of types correspond to different readings of a whole expression. E.g., an unbracketed string $\neg_{t\leftarrow t}\ p_t\ []_{t\rightarrow t}$ works out to logical scope order $\neg[]p$ or $[]\neg p$ depending on the order of function application in its categorial derivation. The single-occurrence restriction allows only finitely many possible readings for a given sequent in the Lambek Calculus, but numbers can vary. Van Benthem 1991, Chapter 9, shows how the familiar categorial combinations have unique readings: all their proofs have the same lambda calculus meaning. E.g., the categorial law $A\rightarrow B,\ B\rightarrow C \Rightarrow A\rightarrow C$ behind the penguin example can only express function composition. The book provides a method in terms of finite automata to generate all readings for a given derivable type transition. Still, there is an open problem of precise *counting formulas* for numbers of readings, or alternatively, for the different single-occurrence lambda normal forms living in arbitrary implicational types. Thus, we do not know the precise degree of compositional ambiguity provided by even the basic categorial calculus. Actually, ambiguity seems a feature of natural language, rather than a defect. Presumably, it serves some useful purpose, say, in efficient coding of frequent situations. Parikh 2001 proposes a game-theoretic approach. But so far no conclusive mathematical explanation has been given.

4 Van Benthem 2002 remarks that invariance seems a typically semantic notion, rather than a proof-theoretic one. Nevertheless, the known connections between permutation invariance and type-theoretic provability suggest deeper links.

5 *SELF* is even the *only permutation-invariant Boolean homomorphism* in its type. Van Benthem 1986 shows how the complete Boolean homomorphisms in any type $(a\rightarrow t) \rightarrow (b\rightarrow t)$ correspond one-to-one with all objects living in type $b\rightarrow a$. The correspondence specializes to the permutation-invariant items in both types. Hence, any permutation-invariant object in the type $(e\rightarrow(e\rightarrow t)) \rightarrow (e\rightarrow t)$, or equivalently $(e\bullet e\rightarrow t) \rightarrow (e\rightarrow t)$, corresponds to a permutation-invariant object in type $e\rightarrow e\bullet e$. But of the latter kind, there is clearly only one, viz. the duplicator map $\lambda x_e\bullet <x_e, x_e>$.

6 In Montague semantics, finite type hierarchies interpret categorial languages as follows: categorial slashes turn into function spaces, and categorial product into

Cartesian product. To modalize these hierarchies, we need *two* ternary relations: (a) z is the result $x(y)$ of applying x to y, (b) z is the ordered pair $\langle x, y \rangle$. Thus, the modal language will have two binary modalities $\langle app \rangle$, $\langle pair \rangle$. Apart from this, in can be treated as our general ternary models. Special axioms then impose mutual constraints on the two relations. E.g., an equivalence between $(AxB) \rightarrow C$ and $A \rightarrow (B \rightarrow C)$ would say that $x(y(z)) = x(\langle y, z \rangle)$. This would contract a categorial isomorphism into an identity. No modal completeness theorem is known for these models.

7 This perspective fits well with the grammar modalities used by Moortgat and others on the analogy of linear logic, allowing simple control over syntactic combination without changing the substructural Lambek base (Moortgat 1997).

8 The low complexity of categorial languages is related to that of the 'poor man's modal languages' studied in Spaan 2000. Nevertheless, there are also some unexplained phenomena. As we noted, without constraints on the ternary relation, the first-order translation validates exactly the non-associative Lambek Calculus *NLC*, whose complexity is in *P*. The associative *LC* arises only with special associativity axioms. But this move explodes modal complexity! The modal logic of associative ternary models is *undecidable*, since it can encode word problems. Evidently, its well-chosen categorial fragments like *LC* escape from this fate, retaining low complexity: *NP* or perhaps (an open question) even *P*. What is the general principle?

9 Stability of type assignment along increasing models may improve in two ways. One is via more uniform model-theoretic constructions out of finite investigations, such as *inverse limits*. Then more forms of first-order statements stabilize (van Lambalgen 1995). Another is making *statistical* assumptions about sample sizes at which all relevant grammatical complexity must show up – as with the ‘shallowness hypothesis’ in the categorial learning procedure of Adriaans 1992.

10 Categorial parsing for meaningful tasks typically involves mediating between two realms: surface syntax and semantic models. That is, complex signs are manipulated (Morrill 1994) which combine syntactic and semantic information, including strings, lambda terms and so on. Indeed, we often get a neo-Montagovian baroque in syntax. From a logical point of view, this exemplifies a typical modern trend: meaningful tasks involve *logic combinations*! Building up forms like this is a

Janus affair, looking at language models on one side, but toward hierarchies of type domains like Montague's original models on the other. Given a language model ***M*** and a type model ***N***, the parsing process builds up a *product model* ***MxN*** that can be mapped homomorphically onto ***M*** and ***N*** through left- and right-projection. Barwise & Seligman 1995 defend this construction as a form of informational unification. Products of modal logics have been investigated in Gabbay & Shehtman 1998. They amalgamate models, giving the combined language interpolation properties.

11 Arrow logics for categories have not yet been studied in great detail. All axioms of basic arrow logic are valid, and so are the mentioned triangle principles, making allowance for the fact that inverse is now a partial function on morphisms. But categories really suggest a *two-sorted* modal language, with separate assertions expressing properties of arrows (morphisms) and of states (objects). State modalities can be used to describe constructions such as products. Two-sorted 'dynamic arrow logics' of this kind have been studied in van Benthem 1996, Chapter 8.4.

12 To see the two levels together, consider a Modus Ponens inference from $If_{t\to t}\ John_e\ comes_{e\to t}$, $Mary_e\ leaves_{e\to t}$ and $John_e\ comes_{e\to t}$ to $Mary_e\ leaves_{e\to t}$. This involves categorial implications for the function types of 'comes' and 'leaves', but there is also the explicit implication 'if' of type $t\to t$.

13 Another source of relevant results might be domain constructions in category theory, suitably tied to concrete inferential phenomena. I am happy to leave this task to Jim Lambek and the formidable group of Montreal category theorists!