Grammar and logic*

by

GLYN MORRILL
Universitat Politècnica de Catalunya, Barcelona

THE POINT OF ORIGIN of logic, historically and conceptually, is the idealisation of reasoning in the form of *arguments* comprising premises and conclusions, and the property of *validity* wherein premises in some sense entail, or have as a consequence, conclusions. In this respect there is a straightforward connection between logic and grammar—the description of language—when the logical semantic relations of a language are taken to be part of the structure to be described grammatically.

In this paper we are concerned with a related but more subtle connection between grammar and logic. In this, linguistic expressions are classified by structured categories which of themselves determine the distributional behaviour of the expressions; these categories are *formulas* of a logic, and the properties of a language predicted by a categorisation of words in a lexicon are the logical consequences of this lexical categorisation (there are no non-logical axioms).

In this manner there emerge *logical foundations of language* in much the same way that in computer science there have emerged *logical foundations of computation*. Applications of the methodology of logic such as the latter require us to see logic as a science not just of natural reasoning, but as a more general science of, let us say, the relations between symbols in virtue of their meaning (especially consequence relations). One can then understand logical methodology in respect of the symbols of computational formalisms and the symbols of natural language in a uniform light. But grammar formalisms are also symbolic systems, and it is our concern here to explain how logical methodology extends uniformly to grammar.

The paper is organised into three sections. In the first we relate

* This paper is a much updated and reoriented version of Morrill (1990). For many valuable discussions at the time of origin of this work I thank Guy Barry, Mark Hepple, and Neil Leslie. All errors are my own.

the historical contributions providing the ingredients for logical foundations of language. In the second section we exemplify linguistic applications. In the final section we briefly discuss linguistic precedents, and in what directions further prospects lie.

1. *Historical ingredients*

1.1. *Logic, types, and signs*

The transition from syllogistic to modern logic was made with Frege's logical analysis of *multiple* quantification in the form of predicate calculus. Such formalism is needed for the representation of multiple quantification in the semantics of natural language. But Frege also addressed the question of how semantic analysis itself is to be organised; to him is attributed the *principle of compositionality*, usually expressed as: 'the meaning of an expression is a function of the meanings of its parts, and their mode of composition'. Although evaluation of the principle solicits greater precision, and can be oriented empirically or methodologically, Frege (1892) saw that an issue is already highlighted in respect of examples like (1).

(1) a. It is not the case that the morning star is the evening star.
b. John believes that the morning star is the evening star.

This is that while the truth of falsity of (1a) depends solely on that of the subordinate sentence "the morning star is the evening star", in (1b) it is not the actual denotation, truth or falsity, of the subordinate sentence which is relevant to the truth of the sentence overall, but rather the idea it expresses (and John's attitude towards that idea): what Frege called the mode of referring or *sense* ("Sinn"). Frege construed sense non-mentalistically, independent of any psychological vehicle. Such issues are addressed in *intensional* semantics; the proper articulation of sense remains an important open question.

Another influence arose through Russell's discovery of inconsistency in Frege's treatment of classes. (There are classes which belong to themselves, $A(A)$, for example the class of all classes; and there are classes which do not belong to themselves, for example the empty class; let B be the class of all classes which do not belong to themselves; then by definition, $\forall X[B(X) \leftrightarrow \neg X(X)]$; but this entails by universal instantiation $B(B) \leftrightarrow \neg B(B)$: contradiction.) Russell responded to such paradoxes by developing a theory of types which stratifies classes. Higher order logical languages, that is ones generalising over propositions and properties, usually need to adopt a version of type theory to avoid inconsistency; such is the case in applications of logic to the semantics of natural language.

As Frege is considered the founder of modern logic, so de Saussure (1916) is considered the founder of modern linguistics. He provided a general semiotic conceptual framework according to which the elements of language are signs comprising a *signifier* (symbol or form) and a *signified* (meaning); in the case of natural language we shall refer respectively to *prosodics* and *semantics*. By distinguishing tokens of language (utterances) from the system itself, he identified language qua a collection of signs as an object of study.

1.2. *Proofs*

The deductive formalisms which we shall employ in relating logic and grammar are *natural deduction* and *sequent calculus*, both due to Gentzen (1934). In natural deduction, proofs are tree-like structures generated by rules from elementary proofs which are single formulas. Connectives have rules of *use* which show how a premise with that principal connective can be used, and rules of *proof*, which show how a conclusion with that principal connective can be proved. In natural deduction these are referred to as rules of elimination (E) and introduction (I) respectively. Recall for instance that the rules for the implicational connective are as follows.

(2) a. $$\frac{\begin{matrix}\vdots & \vdots\\ A & A \to B\end{matrix}}{B}\to E$$

b. $$\frac{\begin{matrix}\overline{A}^{\,n}\\ \vdots\\ B\end{matrix}}{A \to B}\to I^n$$

In $\to$I one *closes* any A assumptions one likes by coindexation with the rule occurrence. Assumptions that have not been closed are open, and a natural deduction proof shows that its open assumptions entail its conclusion.

In sequent calculus, proofs are trees of sequents which, in general, are of the form $\Gamma \Rightarrow \Delta$ where the *antecedent* Γ and *succedent* Δ

are sequences of formulas, stating that the conjunction of Γ entails the disjunction of Δ. Gentzen provided sequent calculus for both classical and intuitionistic logic; for the latter, sequent succedents have at most one formula (and in the absence of negation, exactly one formula). This ensures that a disjunctive conclusion always derives from a proof of one of its disjuncts, and that proofs are *constructive*. It is the intuitionistic case which interests us here because constructivity in logic corresponds to compositionality in grammar.

One may see, just in virtue of the reflexivity of consequence, that sequents of the form of the identity axiom (3a) are valid (corresponding to the elementary natural deduction proofs); similarly, in view of the transitivity of consequence, the Cut rule (3b) is valid.

(3) a. $A \Rightarrow A$ id   b. $$\frac{\Gamma \Rightarrow A \quad \Delta_1, A, \Delta_2 \Rightarrow B}{\Delta_1, \Gamma, \Delta_2 \Rightarrow B}\text{Cut}$$

A further set (4) of *structural* rules are valid. Weakening W embodies monotonicity; Contraction C and Permutation P derive from the idempotency and commutativity of the conjunction represented by the antecedent commas.

(4) a. $$\frac{\Gamma_1, \Gamma_2 \Rightarrow B}{\Gamma_1, A, \Gamma_2 \Rightarrow B}\text{W}$$

b. $$\frac{\Gamma_1, A, A, \Gamma_2 \Rightarrow B}{\Gamma_1, A, \Gamma_2 \Rightarrow B}\text{C}$$

c. $$\frac{\Gamma_1, A, B, \Gamma_2 \Rightarrow C}{\Gamma_1, B, A, \Gamma_2 \Rightarrow C}\text{P}$$

The logical rules themselves are exemplified by the following, for implication.

(5) a. $$\frac{\Gamma \Rightarrow A \quad \Delta, B \Rightarrow C}{\Delta, \Gamma, A \to B \Rightarrow C}\to L$$

b. $$\frac{A, \Gamma \Rightarrow B}{\Gamma \Rightarrow A \to B}\to R$$

As in natural deduction there are rules of use with the connective in the conclusion antecedent, the left (L) rules, and rules of proof with the connective in the conclusion succedent, the right (R) rules. For each sequent proof of $\Gamma \Rightarrow A$ there is a natural deduction proof with open assumptions Γ and root A. Gentzen's *Hauptsatz* was to show *Cut-elimination*: that every sequent proof has a Cut-free counterpart (a normal form) with the same conclusion sequent. There is also such a notion of normalisation in natural deduction (Prawitz 1965); when introduction and elimination rules are composed they constitute a proof detour which can be removed. For example:

(6)

$$\dfrac{\quad \overset{\vdots}{A}}{B}\rightarrow E \quad \rhd \quad \begin{array}{c}\vdots\\ A\\ \vdots\\ B\end{array}$$

Here the proof of A replaces every assumption of A which is closed by the $\rightarrow$I inference. The Cut-free sequent proofs correspond to the normalised natural deduction proofs.

1.3. *Functions*

A contribution of distinct origin though (as we shall shortly note) common content was made by Church's introduction of the λ-calculus as a model of functions and computation. He defined λ-terms representing functions, and modelled computational evaluation as reduction to normal form. In the simply typed formulation (Church 1940) one recursively generates functional types thus: if τ and τ' are types, $\tau \rightarrow \tau'$ is a type. Then given basic terms (variables and constants) for each type, further terms are formed by *application* of a functional term ϕ of type $\tau \rightarrow \tau'$ to an argument term ψ of type τ, $(\phi\ \psi)$ of type τ', and by *abstraction* of a term ϕ of type τ over a variable x of type τ', $\lambda x\phi$ of type $\tau' \rightarrow \tau$. When composed the two operations can be evaluated, thus there is *β-reduction:*

(7) $(\lambda x\phi\ \psi) \rightsquigarrow \phi[x \leftarrow \psi]$

Here the substitution on the right is to replace by ψ every occurrence of x in ϕ bound by the λx.

1.4. *Categorial grammar*

It was also under the influence of the theory of types that, originating with Husserl's semantic categories, and with the Polish school, especially Leśniewski's grammar of semantic categories, Ajdukiewicz introduced a fractional notation for linguistic classification which is the hallmark of what Bar-Hillel later called 'categorial grammar' (Ajdukiewicz 1935; Bar-Hillel 1953; Bar-Hillel, Gaifman and Shamir 1960).

Categorial grammar implements in a rather direct way the compositional method of linguistic analysis due in essence to Frege. Under such analysis certain not necessarily basic expressions are taken to be the primary bearers of meaning, and other expressions are attributed with meanings in terms of the meanings of the expressions in which they occur. Thus Ajdukiewicz presented a notation for categories as the types of these meanings by analogy with the arithmetic law $\frac{x}{y} \times y = x$ such that while expressions like nominals and sentences are categorised by basic categories N and S, other expressions are classified by fractional categories such as $\frac{S}{N}$ for verb phrase (lacking a nominal to be a sentence), and $\frac{(\frac{S}{N})}{(\frac{S}{N})}$ for verb phrase modifier (lacking a verb phrase to form another verb phrase).

The fractional division of Ajdukiewicz does not indicate word order; Bar-Hillel (1953) refined it into two directional varieties / ('over') and \ ('under') such that $A\backslash B$ and B/A indicate combination with an A to the immediate left and right respectively to form a B. Then a verb phrase in a subject-initial clause will be N\S, and a preverbal verb phrase modifier (N\S)/(N\S). Thus in the categorial calculus **AB** of Ajdukiewicz/Bar-Hillel we find the following two rules.

(8) a. $A, A\backslash B \Rightarrow$ B   b. $B/A, A \Rightarrow B$

These rules clearly resemble Modus Ponens, and the directed divisions can be looked upon as directional implications in a sequent system in which none of the structural rules W, C or P are allowed to apply. This connection with logic was fully established by Lambek (1958). He observed that reading $A\backslash B$ (B/A) as the linguistic category of expressions which concatenate with As to the left (right) to form Bs, the rules of proof (9) are valid. (Strictly speaking, Lambek excluded empty antecedent sequents, a detail we shall ignore.)

(9) a. $\dfrac{\Gamma, A, \Rightarrow B}{\Gamma \Rightarrow B/A}/\mathrm{R}$ b. $\dfrac{A, \Gamma \Rightarrow B}{\Gamma \Rightarrow A\backslash B}\backslash\mathrm{R}$

Lambek added an additional, conjunctive, type-constructor 'product' (analogous to arithmetic multiplication) such that **A•B** stands for the immediate succession A and then B, with the following sequent rules of use and proof.

(10) a. $\dfrac{\Gamma_1, A, B, \Gamma_2 \Rightarrow C}{\Gamma_1, A\bullet B, \Gamma_2 \Rightarrow C}\bullet\mathrm{L}$ b. $\dfrac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow A\bullet B}\bullet\mathrm{R}$

And he showed that with the implicational rules of use formulated as in (11), the sequent calculus **L** comprising (3), (9), (10) and (11) enjoys Cut-elimination. (The proof is simpler than Gentzen's precisely because of the absence of structural rules.)

(11) a. $\dfrac{\Gamma \Rightarrow A \quad \Delta_1, B, \Delta_2 \Rightarrow C}{\Delta_1, \Gamma, A\backslash B, \Delta_2 \Rightarrow C}\backslash\mathrm{L}$ b. $\dfrac{\Gamma \Rightarrow A \quad \Delta_1, B, \Delta_2 \Rightarrow C}{\Delta_1, B/A, \Gamma, \Delta_2 \Rightarrow C}/\mathrm{L}$

This has decidability of **L** as a corollary, since in all rules except Cut, the number of connective occurrences in the premises is one less than that in the conclusion. (The same corollary does not follow directly in Gentzen's system since Contraction is complexity-increasing.)

Right at the start of contemporary generative grammar was a presentation of rewriting systems as the paradigmatic mathematical framework for grammar (Chomsky 1957). One of Chomsky's inaugural arguments was that context-free rewriting systems are inadequate for description of natural language, and in this way he motivated the more powerful transformational grammars. The 'classical' categorial grammar **AB** of Ajdukiewicz and Bar-Hillel was shown to be equal in weak generative capacity to context-free phrase structure grammar (Bar-Hillel, Gaifman and Shamir 1960). Chomsky's conjecture (confirmed only recently) that the equivalence also held for Lambek categorial grammar attributed it with the same supposed inadequacy as context-free grammar; see Buszkowski (1997).

1.5. *Formulas-as-types and proofs-as-programs*

The discovery of the connection between λ-calculus and proofs is attributed to Curry. This is that, for example, the implicational formulas of intuitionistic logic can be seen as the types of λ-calculus ('formulas-as-types'), and the natural deduction proofs as the terms ('proofs-as-programs'). Then λ-reduction, cf. (7), and proof normalisation, cf. (6), are the same operation, differing in notation only. That this formulas-as-types correspondence extends to other logical connectives and computational operations was observed by Howard (1969). For example, just as implication corresponds to functional abstraction and application, conjunction corresponds to pairing and projection, and its counterpart to β-reduction is the following:

(12) a. $\pi_1(\phi, \psi) \rightsquigarrow \phi$ b. $\pi_2(\phi, \psi) \rightsquigarrow \psi$

The programming language Lisp of McCarthy et al. (1962) was inspired by λ-calculus in respect of such constructs as LAMBDA (x) and APPLY, and computation as λ-reduction, and thus realised program evaluation as proof normalisation, though without explicit types. It is interesting to note that the construct CONS of Lisp corresponds to pairing and CAR and CDR to first and second projections, i.e. the computational realisations of conjunctive types. For a contemporary account of the Curry-Howard correspondence see Girard, Lafont and Taylor (1989).

1.6. *Grammar and logical semantics*

A language is a medium for the expression of information: a convention for the attribution of meanings to forms. A grammar is a description of a language. We can discriminate in a natural language various notions of form and meaning, at various levels: phonetic and orthographic forms, literal and metaphorical meanings, word, sentence and discourse levels, etc. A grammar might model various aspects of such linguistic reality; we take as our concern the signs comprising the written forms and logical or truth-conditional significance of sentences. A formal grammar is to be a specification of the relation between forms and meanings which is symbolised as a formal (i.e. mechanical) system. The broad structure of a language model set up by a formal grammar will be a pairing of forms with meanings to make signs, and a classification of these signs into categories: sentences, verb phrases, adjectives and so on.

Such a formal description must employ representations of forms, meanings, and categories. Let us represent orthographic forms by sequences of word-forms, and logical meanings by terms of higher order logic. Let $\alpha - \phi$ stand for the sign comprising association of form α and meaning ϕ; let the type assignment statement $\alpha - \phi$: A stand for the assertion that the sign $\alpha - \phi$ is of category A. We call a classification of signs into categories an inhabitation.

Montague (1973) used the λ-calculus and higher order (intensional) logic within a particularly simple architecture for specifying grammar and logical semantics of a language. A lexical component presents an initial inhabitation by word signs, and a rule component presents a set of operations under which the lexical inhabitation is to be closed to generate a general inhabitation of phrasal signs, by pairing the suboperations which apply to form and meaning parts of signs in a rule-by-rule fashion.This parallel structure contrasts with the traditional serial component-to-component architectures where syntactic analysis is seen as feeding into a subsequent stage of semantic analysis.

Consider the following examples of lexical assignments:

(13)	**john**	$-\ \lambda x(x\ \mathbf{j})$	: NP
	mary	$-\ \lambda x(x\ \mathbf{m})$	: NP
	walks	$-$ **walk**	: VP
	sings	$-$ **sing**	: VP
	man	$-$ **man**	: CN
	woman	$-$ **woman**	: CN
	some	$-\ \lambda y\lambda z\exists x[(y\ x) \wedge (z\ x)]$	: D
	every	$-\ \lambda y\lambda z\forall x[(y\ x) \rightarrow (z\ x)]$	: D
	socialist	$-\ \lambda u\lambda v[(\mathbf{socialist}\ v) \wedge (u\ v)]$	: ADJ
	deaf	$-\ \lambda u\lambda v[(\mathbf{deaf}\ v) \wedge (u\ v)]$	: ADJ

Complex expressions can be generated by the following rules; the annotations indicate the compositional operations involved. A statement $\Gamma \Rightarrow X$ asserts that the inhabitants in Γ entail that in X.

(14) $a - x : \mathrm{NP},\ b - y : \mathrm{VP} \Rightarrow a\ b - (x\ y)\text{: S}$ r1
$a - x : \mathrm{D},\ b - y : \mathrm{CN} \Rightarrow a\ b - (x\ y)\text{: NP}$ r2
$a - x : \mathrm{S},\ b - y : \mathrm{S} \Rightarrow a\ or\ b - (x \vee y)\text{: S}$ r3
$a - x : \mathrm{ADJ},\ b - y : \mathrm{CN} \Rightarrow a\ b - (x\ y)\text{: CN}$ r4

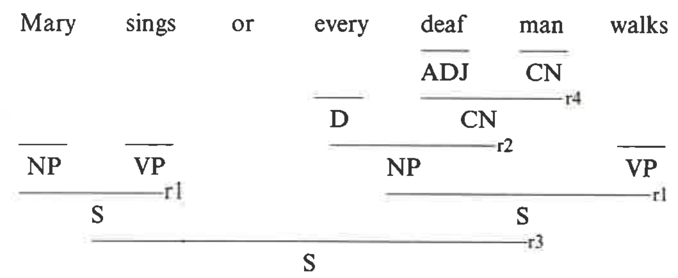


Figure 1: Montague-style derivation

Thus we obtain the analysis in figure 1 of 'Mary sings or every man walks', where the prosodics is left implicit as left-to-right concatenation; the (normalised) semantics of each node is shown in figure 2. Note how in (13) the semantics of proper names has been lifted so that r1 can combine subject proper names and quantifier phrases with verb phrases with the same orientation of application in the semantic operation. There is a trade off, then, in which one reduces the number of syntactic rules at the price of some lexical complexity. Incidentally, a shared category of noun phrase is implied by coordination of the form 'John and every/some woman'.

Montague defined categories categorially, but made no essential use of this, simply stipulating specific rule instances like those in (14) in a phrase structure style. But recalling the origins of categorial grammar, the combination of a functor ($A\backslash B$ or B/A) with its argument (A) is meant to be interpreted by functional application semantically, and it is straightforward to reconstruct an analysis such as the one just given in a purely categorial manner.

We shall assume basic types N (proper name or nominal), CN (common noun), PP (prepositional phrase) and S (sentence) and the following rule schemata (the syntactic operation is left implicit as left-to-right concatenation):

(15) a. x: A, z: $A\backslash B \Rightarrow (z\ x)$: B \E
a. z: B/A, x: $A \Rightarrow (z\ x)$: B /E

Corresponding to the earlier grammar there is the following lexicon. The semantics of proper names has been deflated, and coordination is introduced categorematically, i.e. by lexical assignment rather than syntactic rule.

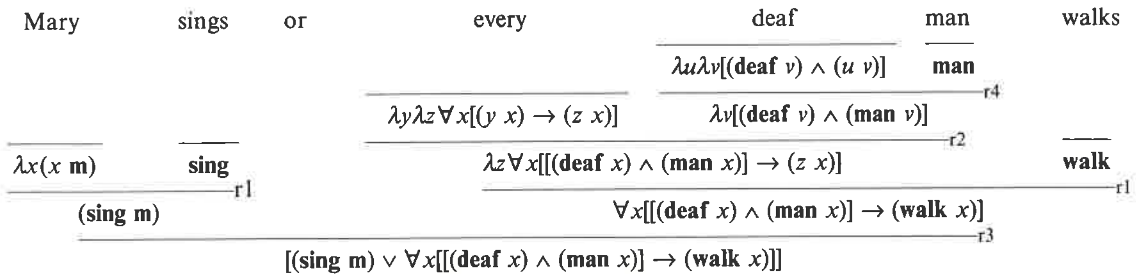


Figure 2: Semantics of Montague-style derivation

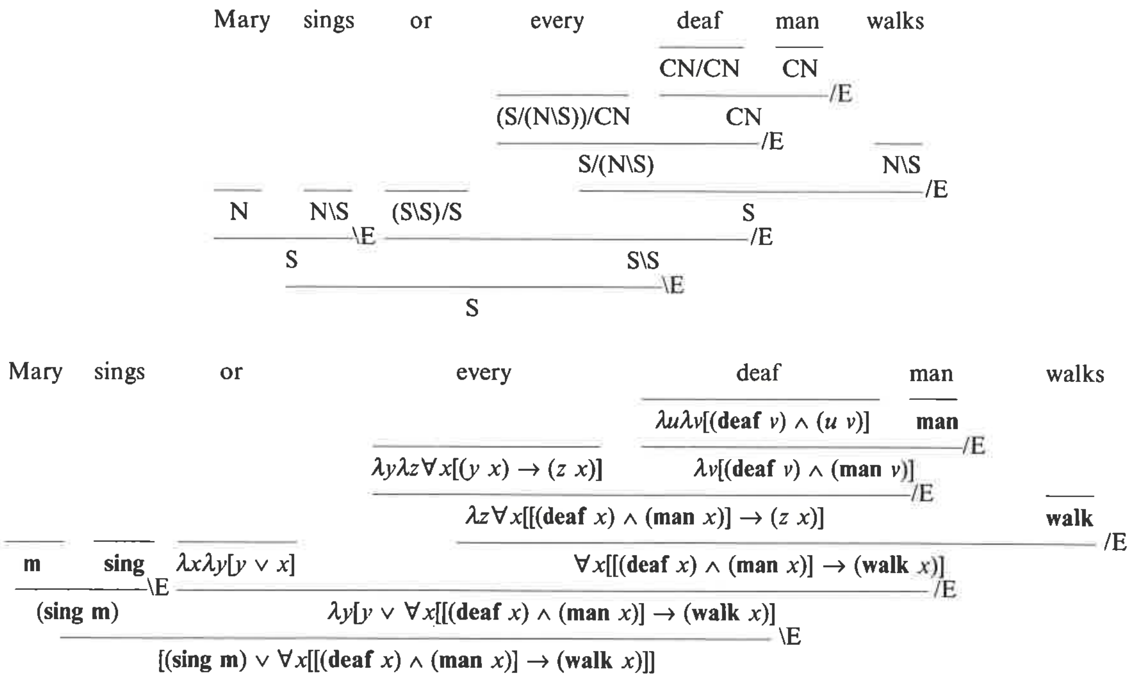


Figure 3: Categorial derivation

(16)	**john**	– **j**	: N
	mary	– **m**	: N
	walks	– **walk**	: N\S
	sings	– **sing**	: N\S
	man	– **man**	: CN
	woman	– **woman**	: CN
	some	– $\lambda y \lambda z \exists x[(y\ x) \wedge (z\ x)]$	: (S/(N\S))/CN
	every	– $\lambda y \lambda z \forall x[(y\ x) \rightarrow (z\ x)]$	: (S/(N\S))/CN
	or	– $\lambda x \lambda y[y \vee x]$	: (S\S)/S
	socialist	– $\lambda u \lambda v[(\mathbf{socialist}\ v) \wedge (u\ v)]$	: CN/CN
	deaf	– $\lambda u \lambda v[(\mathbf{deaf}\ v) \wedge (u\ v)]$	: CN/CN

Now 'Mary sings or every man walks' has the derivation and node-by-node semantic analysis given in figure 3.

This semantic interpretation is extended from **AB** to **L** in the light of the following observations (van Benthem 1983). Categorial logic is a refinement of intuitionistic logic obtained by restricting structural rules. So categorial validity entails intuitionistic validity and for every categorial proof there is an intuitionistic natural deduction proof or typed λ-term obtained by forgetting the structural refinements. In particular, while the rules of use of **L** are semantically interpreted as functional application, the rules of proof are semantically interpreted as functional abstraction. In general, the linguistic *mode of composition* is identified with the logical *constructive content* of the inference. Linguistic ramification of this framework was developed in Moortgat (1988).

1.7. *Linear logic*

The principal exemplar of *substructural logic*, logic lacking some structural rules, is *linear logic*, introduced by Girard (1987). In this, the free application of Contraction and Weakening is withdrawn, thus we obtain an *occurrence* logic in which data (formulas) are considered as resources the presence and number of occurrences of which are significant. Linear logic has both classical and intuitionistic varieties according to whether or not the sequent succedent may be multiple. Van Benthem (1983) in fact works with what can retrospectively be recognised as the so-called multiplicative fragment of intuitionistic linear logic: the calculus **L** plus the permutation rule

(4c), so that the two categorial implications A\B and B/A collapse (into the linear implication $A \multimap B$) and the product $A \bullet B$ becomes commutative (in linear logic, $A \otimes B$: tensor product, or multiplicative conjunction).

Just as implication splits into distinct varieties on withdrawal of Permutation, so linear logic already contains two conjunctions: as well as the multiplicative conjunction $A \otimes B$ one finds the additive conjunction A&B for which the (intuitionistic) rules are:

(17) a. $$\frac{\Gamma_1, A, \Gamma_2 \Rightarrow C}{\Gamma_1, A\&B, \Gamma_2 \Rightarrow C}\&L_a \qquad \frac{\Gamma_1, B, \Gamma_2 \Rightarrow C}{\Gamma_1, A\&B, \Gamma_2 \Rightarrow C}\&L_b$$

b. $$\frac{\Gamma \Rightarrow A \quad \Gamma \Rightarrow B}{\Gamma \Rightarrow A\&B}\&R$$

In the presence of Contraction and Weakening the additive and multiplicative conjunctions collapse. Multiplicative and additive varieties of disjunction also exist, of which the latter, $A \oplus B$, is as follows (the former does not make itself evident given the intuitionistic constraint on succedents):

(18) a. $$\frac{\Gamma_1, A, \Gamma_2 \Rightarrow C \quad \Gamma_1, B, \Gamma_2 \Rightarrow C}{\Gamma_1, A \oplus B, \Gamma_2 \Rightarrow C}\oplus L$$

b. $$\frac{\Gamma \Rightarrow A}{\Gamma \Rightarrow A \oplus B}\oplus R_a \qquad \frac{\Gamma \Rightarrow B}{\Gamma \Rightarrow A \oplus B}\oplus R_b$$

Standard linear logic includes Permutation, but since language is ordered in time we are interested, in grammar, in non-commutativity. Withdrawing Permutation gives *sequence* logics like **L**. The system **L** plus (17) and (18) constitutes the multiplicative and additive fragment of intuitionistic non-commutative linear logic.

Girard reintroduces Weakening and Contraction via a further class of unary connectives, the exponentials (or as we shall say, structural modalities) which licence the structural operations on formulas of which they are the principal connective. In the non-commutative context, (Abrusci 1990) uses the connective 'of course' as follows, such that $!A$ represents discardable, reusable, and permutable As.

(19) a. $\dfrac{\Gamma_1, A, \Gamma_2 \Rightarrow C}{\Gamma_1, !A, \Gamma_2 \Rightarrow B}!L$ b. $\dfrac{!\Gamma \Rightarrow A}{!\Gamma \Rightarrow !A}!R$

c. $\dfrac{\Gamma_1, \Gamma_2 \Rightarrow B}{\Gamma_1, !A, \Gamma_2 \Rightarrow B}!W$

d. $\dfrac{\Gamma_1, !A, !A, \Gamma_2 \Rightarrow B}{\Gamma_1, !A, \Gamma_2 \Rightarrow B}!C$

e. $\dfrac{\Gamma_1, A, B, \Gamma_2 \Rightarrow B}{\Gamma_1, B, A, \Gamma_2 \Rightarrow B}!P$, A or B !-ed

In (19b), $!\Gamma$ stands for a sequence of !-ed formulas, i.e. formulas with principal connective !. The rules !L and !R alone define an **S4** universal modality; !W, !C and !P are structural rules conditioned on the licencing structural modality.

Finally in this section, let us note that it is natural to seek a finer-grained notion of structural modality, as in Barry, Hepple, Leslie and Morrill (1991). Thus for example an **S4** structural modality Δ may be commissioned for just permutation:

(20) a. $\dfrac{\Gamma_1, A, \Gamma_2 \Rightarrow B}{\Gamma_1, \Delta A, \Gamma_2 \Rightarrow B}\Delta L$ b. $\dfrac{\Delta\Gamma \Rightarrow A}{\Delta\Gamma \Rightarrow \Delta A}\Delta L$

$\dfrac{\Gamma_1, A, B, \Gamma_2 \Rightarrow B}{\Gamma_1, B, A, \Gamma_2 \Rightarrow B}\Delta P$, A or B Δ-ed

We shall see in the next section linguistic applications of this and other aspects of non-commutative linear logic in conjunction with the Curry-Howard correspondence.

2. *Type Logical Grammar*

The grammatical architecture exemplified in Montague Grammar is one which sees language objects as having dimensions in syntactic and semantic algebraic domains, and in which rules of grammar correlate operations in these algebras. For this design in general, a sign is an array of elements in the linguistic domains under consideration, a rule is an array of operations; a language is described by the componentwise closure of the lexical signs under the rules of grammar. (For discussion see Oehrle 1988.)

Bidirectional categorial grammar provides for the classification of linguistic objects by starting with primitive types representing "complete" (or: primarily meaningful) expressions, and building further classes by means of the type-constructors / and \. Assignment of a type B/A ($A\backslash B$) to an expression is a simultaneous classification according to form and meaning stating that the expression prefixes (postfixes) itself to expressions of type A to form expressions of type B, and stating that the meaning of the resultant expression is given by the application of the meaning of the affix expression to that of the stem expression.

Consider for instance a language containing as complete expressions some proper names 'John', 'Mary', ... indexed by type N and some sentences 'John walks', 'Mary walks', ..., 'Mary likes John', ... indexed by type S. Then 'walks' has type N\S since it fulfills the condition to have this type, namely to combine with any proper name prefix (and applying semantically as a predicate) to form an S; so also do 'likes John', 'likes Mary', ..., and 'likes' has type (N\S)/N. But additional type assignments are valid under the intended meaning of the type-constructors. The expression 'John' also has type S/(N\S); this is so because given that every verb phrase we may care to choose suffixes to 'John' to form a sentence, it is true that John prefixes to every verb phrase to form a sentence; the semantics in this lifted type is $\lambda x(x\ \mathbf{j})$—the function that applies to the predicate to give the sentence meaning. For similar reasons, 'John' also has type ((N\S)/N)\(N\S), and so on.

The categorial calculus **AB** has an *ordered* natural deduction style presentation as follows:

(21) a. $\dfrac{\begin{matrix}\vdots & & \vdots\\ \chi: B/A & & \phi: A\end{matrix}}{(\chi\ \phi): B}/E$ b. $\dfrac{\begin{matrix}\vdots & & \vdots\\ \phi: A & & \chi: A\backslash B\end{matrix}}{(\chi\ \phi): B}\backslash E$

Here the left-to-right arrangement represents the ordering of assumptions; the encoding of the derivation as an intuitionistic proof in λ-notation (with types implicit) is included before colons.

The calculus **AB** does not capture all the type inferences that

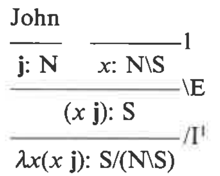


Figure 4: Subject Lifting

are valid according to the interpretation of types described above, e.g. it does not capture that inhabitation of type N by 'John' implies inhabitation also of type S/(N\S). It contains rules of use, but not rules of proof. Completeness for an associative version of affixation (one which forgets grouping) is achieved in the associative Lambek calculus. **L** is obtained in the present natural deduction format by adding the following conditionalisation rules to the directional Modus Ponens rules of **AB**. These state that from a derivation of *B* from assumptions including an (appropriate) occurrence of *A*, a further derivation is obtained by closing that assumption occurrence.

(22) a. $$\dfrac{\begin{array}{c}\overline{x: A}^{\,n} \\ \vdots \\ \psi: B\end{array}}{\lambda x\psi: B/A}/\mathrm{I}^n$$ b. $$\dfrac{\begin{array}{c}\overline{x: A}^{\,n} \\ \vdots \\ \psi: B\end{array}}{\lambda x\psi: A\backslash B}\backslash\mathrm{I}^n$$

There is the condition that *A* is respectively the rightmost and leftmost undischarged assumption above *B* in /I and \I. In the introduction rules, the meaning of the result is given by the functional abstraction over the meaning of the discharged assumption. By way of example, the derivation of figure 4 lifts 'John' of type N with meaning **j** to S/(N\S) with meaning $\lambda x(x\ \mathbf{j})$.

The rules in (22) are analogous to the usual natural deduction rule of → introduction, except that exactly one assumption, in a particular place, must be closed. Derivations are semantically interpreted in a single-bind lambda calculus satisfying the usual law of β-reduction. As observed in the previous section, the functional semantics is an instance of the Curry-Howard correspondence between constructive implicational logic and the lambda calculus. By structuring lexical terms, we may program lexical semantics to interact with derivational semantics under normalisation. By way of

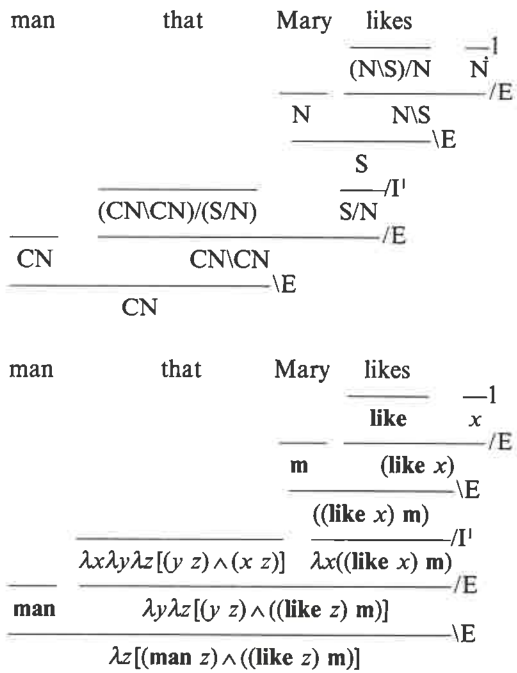


Figure 5: Relativisation

illustration of both this and of hypothetical reasoning we note that the non-canonical constituent 'John likes' can be derived as of type S/N, and assignment of categorial type (CN\CN)/(S/N) to an object relative pronoun produces relativisation (cf. Steedman 1985); see figure 5, where the lexical semantics of the relative pronoun computes the intersection of two predicates.

In general, the depth of embedding of the position that a relative pronoun binds is unbounded. In (23a) it binds the object position in a subordinate clause at depth one, in (23b) at depth two, and the pattern continues without limit.

(23) a. the man that John thinks Mary likes
b. the man that Mary thinks Bill claims Suzy likes
⋮

The treatment just given allows for this long-distance relativisation since there is no constraint on the remoteness of conditionalisation;

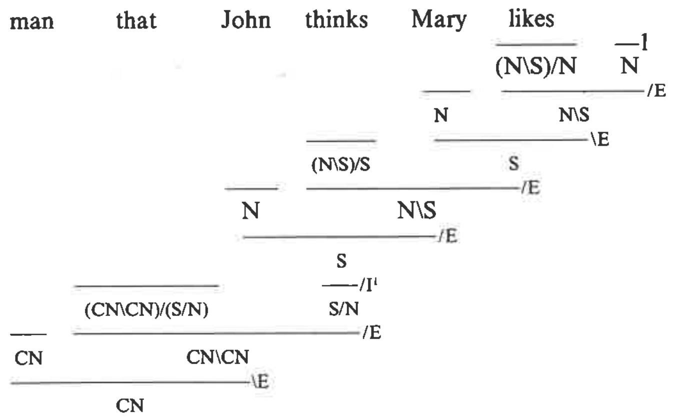


Figure 6: Long distance relativisation

see figure 6. However, the relative pronoun argument clause type S/N admits just clause-final extraction sites ("gaps"): S/N means an element which concatenates with an N at its right periphery to form an S. Thus medial extraction such as e.g. (24) where the object is missing from before the adverb is not generated.

(24) the man that Bill meets today

Furthermore, some verbs in English like 'assure' and 'guarantee' exercise valencies which *must* be satisfied by extraction such as relativisation rather than by a canonical, non-extracted, complement; thus for example (25a) is grammatical but (25b) is not.

(25) a. the man that John assures Mary to be reliable
b. *John assures Mary Bill to be reliable.

We shall later see how the desired effects can be obtained.

For the product types $A \bullet B$ there are the following rules of ordered natural deduction:

(26) a.
$$\frac{\begin{matrix}\vdots & \vdots\\ \phi: A & \psi: B\end{matrix}}{(\phi, \psi): A \bullet B}\bullet\text{I}$$

b.
$$\frac{\begin{matrix}\vdots\\ \dfrac{\omega: A \bullet B}{x: A \quad y: A}\bullet\text{E}^n\\ \vdots\\ \gamma: C\end{matrix}}{\gamma[x \leftarrow \pi_1\omega, y \leftarrow \pi_2\omega]: C}n$$

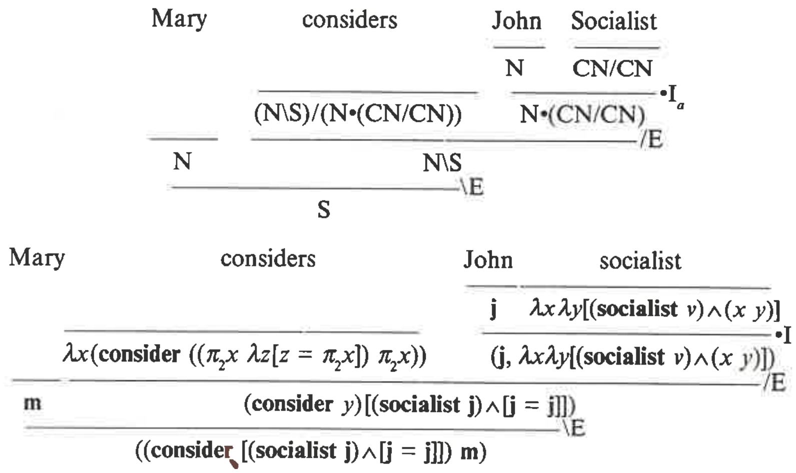


Figure 7: Analysis using product

In the rule $\bullet$E we plug vertically a proof of $A \bullet B$ into the adjacent premises A and B of a proof of C. The root is extended by a coindexed bar in order that the derivations remain freely generated, and the semantic annotation is recoverable from just the formulas and rules.

In the example of figure 7 the product is used to give a so-called "small-clause" analysis to the complements of the verb 'consider' whereby the object and predicative following the verb form a canonical constituent. The lexical semantics of 'considers' yields an analysis [(**socialist j**) ∧ [**j** = **j**]] of the subordinate clause which is logically equivalent to (**socialist j**). (Note that this illustrative role of product is not essential since types $C/(B \bullet A)$ and$(C/A)/B$ are interderivable in **L**.)

The examples above illustrate some potential of Lambek categorial grammar in relation to natural language, even for "action-at-a-distance" such as relativisation where an element has an effect remote from its location, and which classically motivates a transformational approach. But this is clearly only a beginning, for instance natural language abounds with polymorphism, and while the Lambek calculus is a sequence logic suited to the description of linear linguistic forms, it of itself does not offer any control over the subtleties of occurrence and order in natural language. In the following subsections we review one small further step, extending categorial grammar, seen as an implicational fragment of non-commutative linear logic, with type-constructors inspired by linear logic, and applying these to various problematic phenomena.

2.1. *Additives*

The additive conjunction type-constructor & has the following rules of natural deduction.

(27) a.

$$\dfrac{\begin{matrix} \vdots\overline{\Gamma} & \overset{-n}{\Gamma} \\ & \vdots \\ \phi\colon A & \psi\colon B \end{matrix}}{(\phi,\ \psi)\colon A\,\&\,B}\,\&\mathrm{I}^n$$

b.

$$\dfrac{\begin{matrix}\vdots\\ \chi\colon A\,\&\,B\end{matrix}}{\pi_1\chi\colon A}\,\&\mathrm{E}_a \qquad \dfrac{\begin{matrix}\vdots\\ \chi\colon A\,\&\,B\end{matrix}}{\pi_2\chi\colon B}\,\&\mathrm{E}_b$$

In &I, $\overline{\Gamma}$ signifies cancellation of the sequence Γ of leaves.

Consider the ambiguity of the following, between creating a picture, and coating the building.

(28) John paints the house.

The relation between the meanings is such that ‘paints’ is a transitive verb in each case. The assignment in (29) captures this generalisation.

(29) **paints** – $\lambda x \lambda y(((\mathbf{paint}_1\ x)\ y),\ ((\mathbf{paint}_2\ x)\ y))$: (N\(S&S))/N

Depending on the final step, figure 8 delivers one or other of the two readings.

For the additive disjunction ⊕ the rules of natural deduction are thus:

$$\dfrac{\dfrac{\text{S\&S}}{\text{S}}\,\&\mathrm{E}_a \text{ or } \&\mathrm{E}_b}{} \quad\text{from}\quad \dfrac{\dfrac{\text{John}}{\text{N}} \quad \dfrac{\dfrac{\text{paints}}{\text{(N\\(S\&S))/N}} \quad \dfrac{\text{the house}}{\text{N}}}{\text{N\\(S\&S)}}/\mathrm{E}}{\text{S\&S}}\backslash\mathrm{E}$$

Figure 8: Polymorphism through conjunction

(30) a.

$$\dfrac{\begin{matrix}\vdots\\ \phi\colon A\end{matrix}}{\iota_1\phi\colon A\oplus B}\oplus\mathrm{I}_a \qquad \dfrac{\begin{matrix}\vdots\\ \psi\colon B\end{matrix}}{\iota_2\psi\colon A\oplus B}\oplus\mathrm{I}_b$$

b.

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ \omega\colon A\oplus B\end{matrix}}{\Gamma\quad x\colon A\ \Delta \qquad \Gamma\ y\colon B\quad \Delta}\oplus\mathrm{E}^n \\ \begin{matrix}\vdots & & \vdots\\ \chi_1\colon C & & \chi_2\colon C\end{matrix}}{(\omega \to x.\chi_1;y.\chi_2)\colon C}\,n$$

In ⊕E a subproof of $A\oplus B$ is plugged into the premises A, Δ and Γ, B of two proofs of C. The semantic operator is a case operator keyed on first and second injections ι_1 and ι_2. Depending on the labelling, the first or second branch is taken:

(31) $(\iota_1\phi \to x.\chi_1;\ y.\chi_2) \rightsquigarrow \chi_1[x\leftarrow\phi]$
$(\iota_2\psi \to x.\chi_1;\ y.\chi_2) \rightsquigarrow \chi_2[y\leftarrow\psi]$

This disjunctive connective finds application in e.g. the polymorphism of ‘is’ with respect to its complementation:

(32) a. 007 is teetotal.
b. 007 is Bond.

In (32a) the complement is an adjective and the semantics is predication of the subject; in (32b) the complement is a nominal and the semantics involved is identification with the subject. We achieve these effects with the following single assignment.

(33) **is** – $\lambda x \lambda y(x \to z.((z\ \lambda u[u = y])\ y);\ w.[w = y])$
: (N\S)/((CN/CN)⊕N)

A derivation illustrating the predicational use is given in figure 9. A derivation for the identificational use is shown in figure 10. The use of such type-constructors, as well as quantifiers, in application to linguistic polymorphism is discussed in more detail in Morrill (1994a, ch. 6).

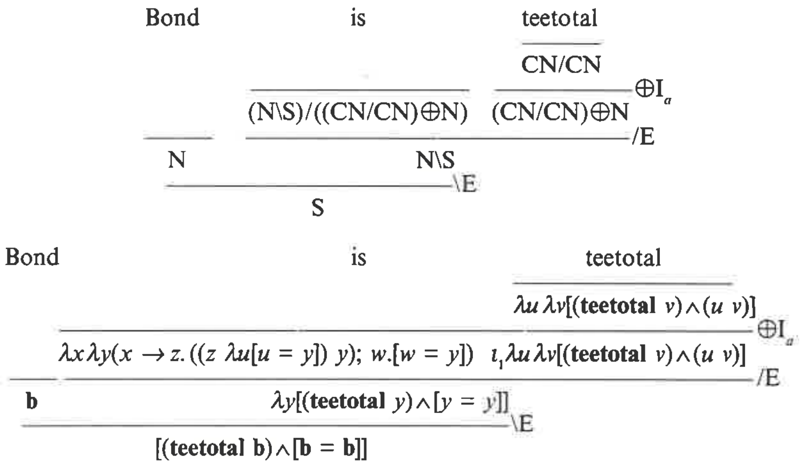


Figure 9: Polymorphism through disjunction, I

2.2. *Semantic Modality*

Intensional semantics provides an approach to the problem of sense on the basis of multiple reference or *contextuality*. In this view meanings are relativised to worlds or points of reference. Montague (1973) structures semantic types as ordered pairs thus: if τ and τ' are types, then $<\tau, \tau'>$, and $<s, \tau>$ are types. The former is the usual binary functional type-constructor of the simple theory of types; the latter is a unary type-constructor for functions into τ from the set of worlds represented by s: senses formalised as intensions. To the definition of lambda terms are added the clauses: if ϕ is a term of type τ then ${}^\wedge\phi$ (intensionalisation; 'up') is a term of type $<s, \tau>$, and if ϕ is a term of type $<s, \tau>$, ${}^\vee\phi$ (extensionalisation; 'down') is a term of type τ. There is the following law of term reduction, *down-up cancellation*:

(34) $\quad {}^{\vee\wedge}\phi \rightsquigarrow \phi$

The intensionalisation and extensionalisation operators are like QUOTE and EVAL in Lisp. Hobbs and Rosenschein (1978) point out the close relations between Lisp and Montague's intensional logic.

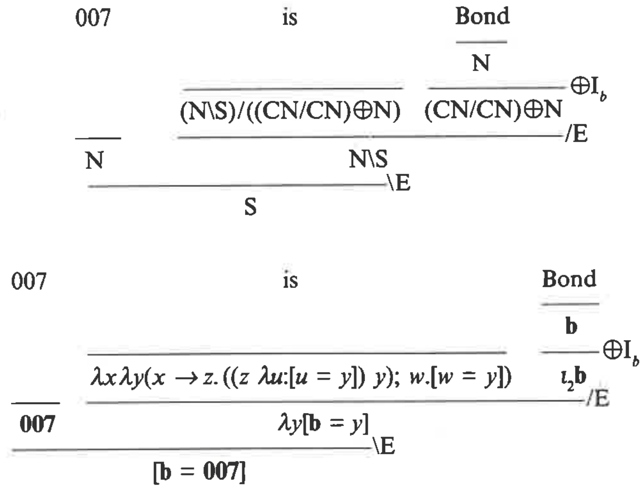


Figure 10: Polymorphism through disjunction, II

In Morrill (1990b) this is rendered as a Curry-Howard correspondence between modal logic and intensional types; the unary modal operator $\Box$ is the intensional type-constructor. Annotated rules of **S4** natural deduction are as follows:

(35) a.
$$\frac{\begin{matrix}\vdots\\ \phi: A\end{matrix}}{{}^\wedge\phi: \Box A}\Box I^*$$

b.
$$\frac{\begin{matrix}\vdots\\ \phi: \Box A\end{matrix}}{{}^\vee\phi: A}\Box E$$

There is the condition * on $\Box$I that every path from the root to an open assumption contains a licencing modal type, i.e. a modal type which does not depend on closed assumptions. An intensionalised version of lexical assignments is as follows.

(36)

for	– $^\wedge\lambda x x$	: $\Box$(PP/N)
john	– $^\wedge$**j**	: $\Box$N
mary	– $^\wedge$**m**	: $\Box$N
likes	– **like**	: $\Box$((N\S)/N)
man	– **man**	: $\Box$CN
the	– **the**	: $\Box$(N/CN)

thinks	– **think**	: □((N\S)/□S)
votes	– **vote**	: □((N\S)/PP)

All lexical categories are modalised since all words have senses. The semantics of the preposition is the identity function at all worlds and that of the proper names renders them *rigid designators*, i.e. they have the same reference at all worlds; the propositional attitude verb 'thinks' projects an intensional subordinate sentence. A derivation is given in figure 11, an intensional analysis of 'Mary thinks John votes for the man'.

This semantic modality rests on an intensional semantic interpretation on the basis of a single domain of points of reference, but e.g. Montague distinguishes worlds and times, and for this we could use a bimodal logic. In general there would be a polymodal logic, with a modality for each dimension of intensional ramification; see Morrill (1994a, ch. 5).

2.3. *Structural Modality*

There is the following annotated natural deduction rule of controlled permutation with **S4** modality introduction and elimination.

(37) a.

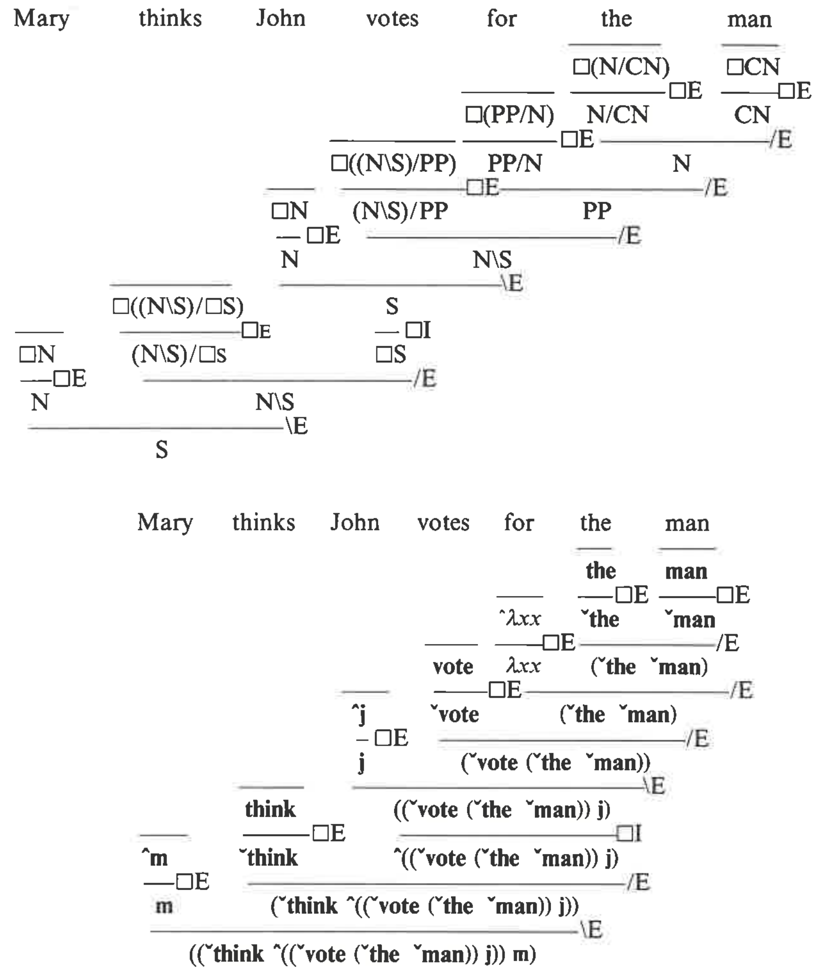
$$\dfrac{\dfrac{\begin{matrix}\vdots & & \vdots\\ \phi\colon A & & \psi\colon B\end{matrix}}{\begin{matrix}y\colon B & & x\colon A\end{matrix}}\,\Delta P^{n},\ A \text{ or } B\ \Delta\text{-ed} \\ \vdots \\ \chi\colon C}{\chi[x\leftarrow\phi,\ y\leftarrow\psi]\colon C}\,n$$

b.

$$\dfrac{\begin{matrix}\vdots\\ \phi\colon A\end{matrix}}{\phi\colon \Delta A}\,\Delta I^{*} \qquad \dfrac{\begin{matrix}\vdots\\ \phi\colon \Delta A\end{matrix}}{\phi\colon A}\,\Delta E$$

Assignment of type (CN\CN)/(S/ΔN) to a relative pronoun will now allow it to fill a gap in *any* position, not just a clause-peripheral one, as seen in figure 12, cf. (24).

Consider now the fact, illustrated in (25), that 'assure' allows extraction of the subject of its infinitival complement, but does not



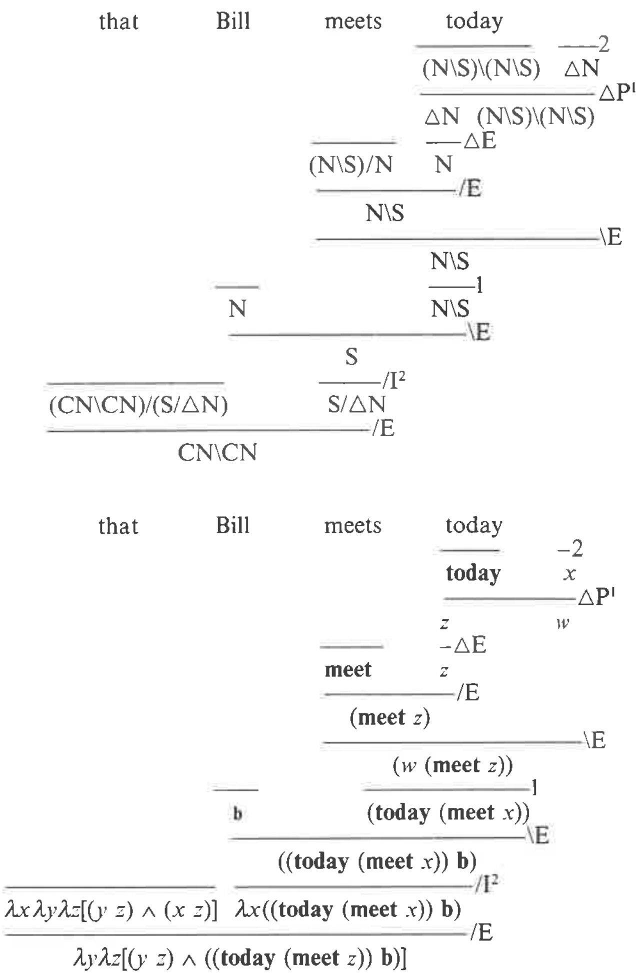


Figure 12: Medial extraction

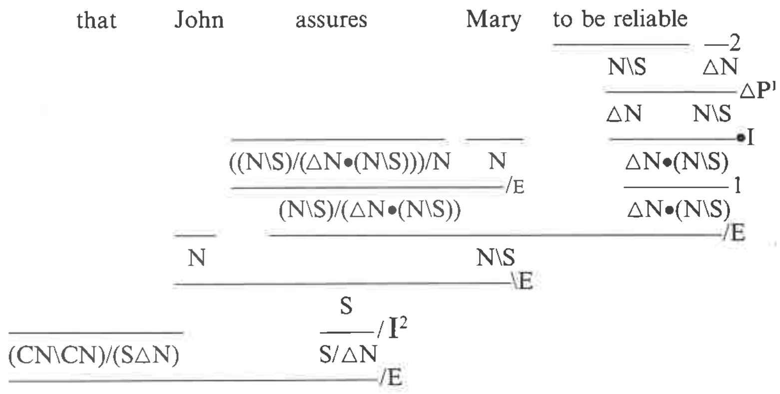


Figure 13: Obligatory extraction

allow a canonical, i.e. non-extracted, form. A category that can only allow extraction is provided by ((N\S)/(△N•(N\S)))/N. Thus, on the one hand (25a) can be derived as shown in figure 13. On the other hand, the sequent (38), corresponding to (25b), is not modally valid (and nor is any variant where the underlined N corresponding to 'Bill' is relocated).

(38) N, ((N\S)/(△N•(N\S)))/N, N, $\underline{\text{N}}$, N\S ⇒ S

The reason why is that N ⇒ △N is not valid. In this way the structural properties of △ allow left extraction, while the modal properties block generation of any canonical form.

3. *Prospects*

In section 1 we have tried to explain the precedents of type logical grammar from the point of view of logic. We offer now a brief analysis from the point of view of formal linguistics.

The model of grammar based on rewriting systems initially established by Chomsky (1965) had the following two fundamental characteristics: firstly, the lexicon was the repository of only *idiosyncratic* information, all generalisations being expressed syntactically;

secondly, surface structures were derived from deep structures by *multistratal* derivations generated by series of *transformations*. Type logical grammar can be seen as the integration of Chomsky's formal syntax and Montague's formal semantics in the light of three subsequent revolutions: one linguistic, one computational, and one logical.

The linguistic revolution is the transfer to lexicalism initiated in Lexical-Functional Grammar (Bresnan 1982), driven by empirical considerations revealing that various phenomena (such as passivisation) are best characterised by lexical generalisations. This move abandons the principle that all regularities should be expressed syntactically and not lexically.

The computational revolution is the elimination of strata. Computationally, an architecture comprising many serially related representations is unattractive because calculations at one point must be made in ignorance of constraints exercised at another; one would prefer all dependencies to represented simultaneously so that computations and constraints can be rendered in parallel. This is realised in such monostratal computational linguistic formalisms as Generalised Phrase Structure Grammar (Gazdar, Klein, Pullum and Sag, 1985) and Head-driven Phrase Structure Grammar (Pollard and Sag 1987, 1994).

The logical revolution (Moortgat 1988, van Benthem 1991, Morrill 1994a) marks a transition from grammar as formal system in general, as in Chomsky's rewriting paradigm, to logical system in particular, in the manner we have here endeavored to explain and illustrate. A type logical grammar comprises solely a lexicon, there are no non-logical axioms such as analogues of phrase structure schemata, so in so far as a logic may be defined model-theoretically, this framework is actually non-stratal: it makes no essential use of any level of representation, rather the language model is determined entirely be the meaning of operators. It is intriguing to note that the most recent theory *minimalism* of Chomsky (1995) shares the same features of lexicalism and non-existence of any essential level of syntactic representation (see Morrill, 1994a, ch. 9). Thus it does not seem in vain to hope for a methodological convergence here.

Let us conclude with two observations as to future prospects, one at the level of category formulas, the other at the level of proofs.

Regarding categorisation, we have assumed above a single sort of prosodic form, strings of words, and prosodic operation, concatenation. That is, we have worked with a system which drops permutation but retains the associativity implicit in the definition of sequent antecedents as sequences. Lambek (1961) already introduced non-associative logic. However, rather than assuming full associativity or full non-associativity, what really seems preferable is some kind of controlled partial non-associativity, introducing localised structural inhibition analogous to the structural facilitation of controlled partial commutativity seen above (Morrill 1994b, 1994a ch. 7).

Still, this partial commutativity itself is not adequate for action-at-a-distance in general and the subtleties of discontinuity in natural grammar demand a methodology for combining concatenative adjunction with other modes such as interpolation to introduce partial commutativity (Moortgat 1991, Morrill 1994a, chs. 4–5,1995, Morrill and Merenciano 1997). A general framework for such *multimodal* categorial type logic is given in Moortgat (1997).

Concerning proofs, we have employed for purposes of exposition a version of ordered natural deduction, this resembling the derivations seen in linguistic contexts; we have also spoken of sequent calculus. As just remarked, the manner in which derivations are represented is not linguistically relevant, being just the notation for calculations as to what the grammar says. However, computationally one is certainly interested in the representation of grammatical information and inference, and here it is important that a proof syntax, proof nets (Girard 1987, Danos and Regnier 1989), has been developed which can make a real claim to represent the deep structure behind different notations for proofs, and to define the global geometric conditions of interaction that determine the validity of proofs.

Entering into just a little detail, let us note that although natural deduction proofs represent a degree of parallelism of derivation (independent subproofs may be performed in parallel), they still exercise a seriality in respect of hypothetical reasoning: assumptions are made available and at some point discharged. Ideally we would rid ourselves of all such temporal clutter and identify the pure structure, i.e. the geometry, of proofs, in such a way that parallelism, dependence, and independence are fully determined. This is what is done in proof nets.

In the linguistic, non-commutative, form (Roorda 1991, Abrusci 1995) this offers us the potential for a *geometry of language* in which proof nets provide us with the true deep structure of linguistic derivations. Furthermore, an understanding of proof nets may begin to explain computational processes under the Curry-Howard paradigm. In particular, it may help connect *evaluation* with *mode of referring*, i.e. sense. Intensional semantics treats intensions as functions-in-extension from contexts to denotations; hence the indistinguishability of logical equivalents and the problems of logical omniscience in Montague semantics; the treatment needs to be more fine-grained and less extensional. On the other hand, the analogue arising in Lisp is too quotational, identifying in effect sense with form, which is too fine-grained. Frege sought a non-psychological notion of sense; but information processing, human or machine, as we understand it today is surely at the centre of the issue, and a deeper understanding of proofs and proofs-as-programs may provide us with a key.

References

ABRUSCI, V.M.: 1990, 'Noncommutative intuitionistic linear propositional logic', *Zeit. f. Math.Log. u. Grund. d. Math.*

ABRUSCI, V.M.: 1995, 'Noncommutative proof nets', in Jean-Yves GIRARD, Yves LAFONT and Laurent REGNIER (eds.), *Advances in Linear Logic*, London Mathematical Society Lecture Note Series 222, Cambridge University Press, Cambridge, 271–296.

AJDUKIEWICZ, Kazimierz: 1935, 'Die syntaktische Konnexität', *Studia Philosophica* **1**, 1–27. Translated in S. MCCALL: 1967 (ed.), *Polish Logic: 1920–1939*, Oxford University Press, Oxford, 207–231.

BAR-HILLEL, Y.: 1953, 'A quasi-arithmetical notation for syntactic description', *Language* **29**, 47–58.

BAR-HILLEL, Y., C. GAIFMAN, and E. SHAMIR: 1960, 'On Categorial and Phrase Structure Grammars', *Bull. Res. Council Israel* F 9, 1–16.

BARRY, Guy, Mark HEPPLE, Neil LESLIE, and Glyn MORRILL: 1991, 'Proof Figures and Structural Operators for Categorial Grammar', in *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics*, 198–203.

VAN BENTHEM, Johan: 1983, 'The semantics of Variety in Categorial Grammar', Report 83-29, Department of Mathematics, Simon Fraser University. In W. BUSZKOWSKI, W. MARCISZEWSKI, and J. VAN BENTHEM: 1988 (eds.), *Categorial Grammar*, Volume 25, Linguistic & Literary Studies in Eastern Europe, John Benjamins, Amsterdam, 37–55.

VAN BENTHEM, Johan: 1991, *Language in Action: Categories, Lambdas and Dynamic Logic*, Studies in Logic, North-Holland, Amsterdam.

BRESNAN, Joan W.: 1982 (ed.), *The Mental Representation of Grammatical Relations*, MIT Press, Cambridge, Massachusetts.

BUSZKOWSKI, Wojciech: 1997, 'Mathematical linguistics and proof theory', in J. VAN BENTHEM and A. TER MEULEN (eds.), *Handbook of Logic and Language*, Elsevier, Amsterdam, 683–736.

CHOMSKY, Noam: 1957, *Syntactic Structures*, Mouton, The Hague.

CHOMSKY, Noam: 1965, *Aspects of the Theory of Syntax*, MIT Press, Cambridge, Massachusetts.

CHOMSKY, Noam: 1995, *The Minimalist Program*, MIT Press, Cambridge, Massachusetts.

CHURCH, Alonzo.: 1940, 'A formulation of a simple theory of types', *Journal of Symbolic Logic* **5**, 56–68.

DANOS, Vincent and Laurent REGNIER: 1989, 'The structure of multiplicatives', *Archives for Mathematical Logic* **28**, 181–203.

FREGE, Gottlob: 1892, 'Über Sinn und Bedeutung', *Zeitschrift für Philosophie und philosophische Kritik* **100**, 25–50. Translated as 'On sense and reference' in P.T. GEACH and M. BLACK: 1960 (eds.), *Translations from the Philosophical Writings of Gottlob Frege*, Blackwell, Oxford, 56–78.

GAZDAR, Gerald, Ewan KLEIN, Geoffrey PULLUM, and Ivan SAG: 1985, *Generalised Phrase Structure Grammar*, Blackwell, Oxford.

GENTZEN, G.: 1934, 'Untersuchungen über das logische Schliessen', *Math. Zeitschrift* **39**, 176–210, 405–431. Translated as 'Investigations into Logical Deduction' in M. SZABO: 1969 (ed.), *The Collected Papers of Gerhard Gentzen*, North-Holland, Amsterdam, 68–131.

GIRARD, Jean-Yves: 1987, 'Linear Logic', *Theoretical Computer Science* **50**, 1–102.

GIRARD, Jean-Yves, Yves LAFONT, and Paul TAYLOR: 1989, *Proofs and Types*, Volume 7, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, Cambridge.

HOBBS, Jerry R. and Stanley J. ROSENSCHEIN: 1978, 'Making Computational Sense of Montague's Intensional Logic', *Artificial Intelligence* **9**, 287–306.

HOWARD, W.A.: 1969, 'The formulae-as-types notion of construction', ms. In J.R. HINDLEY and J.P. SELDIN: 1980 (eds.), *To H.B. Curry, Essays on Combinatory Logic, Lambda Calculus and Formalism*, Academic Press, New York.

LAMBEK, J.: 1958, 'The mathematics of sentence structure', *American Mathematical Monthly* **65**, 154–170. Also in W. BUSZKOWSKI, W. MARCISZEWSKI, and J. VAN BENTHEM: 1988 (eds.), *Categorial Grammar*, Linguistic & Literary Studies in Eastern Europe Volume 25, John Benjamins, Amsterdam, 153–172.

LAMBEK, J.: 1961, 'On the calculus of syntactic types', in R. JAKOBSON (ed.), *Structure of language and its mathematical aspects*, Proceedings of the Symposia in Applied Mathematics **XII**, American Mathematical Society, Rhode Island, 166—178.

McCarthy, J., et al.: 1962, *LISP 1.5 Programmer's Manual*, MIT Press, Cambridge, Massachusetts.

Montague, Richard: 1973, 'The proper treatment of quantification in ordinary English', in J. Hintikka, J.M.E. Moravcsik and P. Suppes (eds.) *Approaches to Natural Language*, D. Reidel, Dordrecht, 221–242. Also in R.H. Thomason: 1974 (ed.), *Formal Philosophy: Selected Papers of Richard Montague*, Yale University Press, New Haven, 247–270.

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

Moortgat, Michael: 1991, 'Generalised Quantification and Discontinuous Type Constructors', in H. Bunt and A. van Horck: 1996 (eds.), *Discontinuous Constituency*, Mouton de Gruyter, Berlin, 181–208.

Moortgat, Michael: 1997, 'Categorial type logics', in J. van Benthem and A. ter Meulen (eds.), *Handbook of Logic and Language*, Elsevier, Amsterdam, 93–177.

Morrill, Glyn: 1990a, 'Grammar and Logical Types', in Martin Stockhof and Leen Torenvliet (eds.), *Proceedings of the Seventh Amsterdam Colloquium, Pt. 2*, Institute for Language, Logic and Information, Amsterdam, 429–450. Also in Guy Barry and Glyn Morrill: 1990 (eds.), *Studies in Categorial Grammar*, Edinburgh Working Papers in Cognitive Science Volume 5, 127–148.

Morrill, Glyn: 1990b, 'Intensionality and Boundedness', *Linguistics and Philosophy* **13**, 699–726.

Morrill, Glyn: 1994a, *Type Logical Grammar: Categorial logic of signs*, Kluwer Academic Publishers, Dordrecht.

Morrill, Glyn: 1994b, 'Structural Facilitation and Structural Inhibition', in M. Abrusci, C. Casadio and M. Moortgat (eds.), *Linear Logic and Lambek Calculus*, Proceedings 1993 Rome Workshop, ESPRIT BRA 6852 DYANA–2, 183–210.

Morrill, Glyn: 1995, 'Discontinuity in Categorial Grammar', *Linguistics and Philosophy* **18**, 175–219.

Morrill, Glyn and Josep-Maria Merenciano: 1997, 'Generalising discontinuity', *Traitement automatique des langues* **27**, 119–143.

Oehrle, Richard T.: 1988, 'Multi-Dimensional Compositional Functions as a Basis for Grammatical Analysis', in R. Oehrle, E. Bach and D. Wheeler (eds.) *Categorial Grammars and Natural Language Structures*, D. Reidel, Dordrecht, 349–389.

Pollard, Carl J. and Ivan A. Sag: 1987, *Information-Based Syntax and Semantics: Volume 1—Fundamentals*, Number 13, CSLI Lecture Notes, Center for the Study of Language and Information, Stanford.

Pollard, Carl J. and Ivan A. Sag: 1994, *Head-driven Phrase Structure Grammar*, Chicago University Press, Chicago.

Prawitz, D.: 1965, *Natural Deduction: A Proof-Theoretical Study*, Almqvist and Wiksell, Uppsala.

Roorda, Dirk: 1991, *Resource Logics: proof-theoretical investigations*, Ph.D. dissertation, Universiteit van Amsterdam.

de Saussure, Ferdinand: 1916, *Cours de linguistique générale*, translated in R. Harris: 1983, *Course in General Linguistics*, Duckworth, London.

Steedman, Mark: 1985, 'Dependency and Coordination in the Grammar of Dutch and English', *Language* **61**, 523–568.