Proofs and Linguistic Categories: Applications of Logic to the Analysis and Implementation of Natural Language. V.M. Abrusci & C. Casadio (eds), Proceedings 1996 Roma Workshop

Head Promotion and Obliqueness in a Multimodal Grammar

Mark Hepple
Department of Computer Science,
University of Sheffield, UK

Abstract

This paper describes a framework for multimodal categorial systems, i.e. systems that allow different modes of logical behaviour to be displayed within a single system. From a linguistic point of view, this characteristic amounts to making available multiple modes of linguistic description within a single formalism. A key advantage is that when it comes to treating a given phenomenon, the multimodal approach allows that a level of description be exploited that encodes only those aspects of linguistic structure that are relevant to the phenomenon. This advantage is illustrated in relation to the analysis of binding phenomena. The treatment of command asymmetries in binding is based on adopting an approach where grammatical hierarchy is encoded via categorial argument order. This treatment of grammatical hierarchy is itself made possible by adapting a proposal for the treatment of English in which observed word order may result from local verb movement, called *head promotion*.

1 Introduction

The theory of grammatical relations proposed in Dowty [3], in which grammatical hierarchy or relative obliqueness is encoded via categorial argument order (i.e. subcategorisation order), has provided the basis for accounts of a range of phenomena within Montague Grammar, including relation changing, control, and command relations in binding. It is well known that this approach is not compatible with a purely concatenative treatment of string formation, requiring in Montague Grammar use of the non-concatenative string operation *wrap* [1]. Of various attempts to reconstruct wrap within alternative frameworks, we note particularly the proposal of Jacobson [8], in which English has an 'underlying' order where verbs follow their direct objects, with observed word order resulting from *head promotion*, i.e. bounded movement of the verb to VP initial position.

In this paper, an approach to formulating multimodal categorial systems is described. Then, it is shown that an argument order based treatment of grammatical hierarchy can be employed within a multimodal grammar, when it is combined with a treatment of word order that is inspired by Jacobson's proposal. This grammar is then applied to the treatment of binding relations. The multimodal approach allows us to choose for any linguistic phenomenon addressed, a level of description that encodes only the aspects of linguistic structure that are relevant to the treatment of that phenomenon. In the case of binding, this allows us to handle the alternative bound uses of each anaphor or personal pronoun whilst avoiding the need for multiple lexical assignments.

2 Multimodal Categorial Grammar

Recent work in Categorial Grammar has seen the proposal of a number of 'multimodal' approaches, which make available multiple modes of construction, realised in syntax via different product operators $\circ_\alpha$ — each with associated implicationals $\overset{\alpha}{\rightarrow}$, $\overset{\alpha}{\leftarrow}$, its left and right residuals.[1] For each product operator there is a corresponding operator in the underlying interpretive semantics, and the behaviour of the former reflects the axioms that govern the latter in the interpretive semantics. Such axioms can be divided into three subgroups: (i) *mode internal axioms*, which involve only a single modality, e.g. familiar associativity $x \circ_i (y \circ_i z) = (x \circ_i y) \circ_i z$; (ii) *interaction axioms*, where more than one modality is involved, e.g. $x \circ_i (y \circ_j z) = (x \circ_i y) \circ_j z$; (iii) *linkage* or *inclusion axioms*, which allow derivation from one mode to another, e.g. $x \circ_i y \longrightarrow x \circ_j y$. Intuitively, the move from one mode to another allowed by a linkage axiom is akin to movement from one description of a linguistic object to another less informative description.

2.1 A Natural Deduction Formulation

We adopt a *labelled* natural deduction formulation, employing inference rules (1–3) below. In this formulation, labelled formulae take the form: $m \vdash A : s$, with A a type, s a 'semantic' lambda term, and m a *marker* term. Marker terms are structured objects that are built up as deduction proceeds, and the structural information they record is used in ensuring appropriate structure sensitivity in deduction (and hence this system is an instance of a *labelled deductive system* [4]). In linguistic derivations, assumptions that correspond to lexical items have lexically provided marker and semantic components (loosely, the word's 'string' or 'phonology' and its meaning). Other assumptions have a simple variable for their marker and semantic terms (which may be given as the *same* variable, to aid readability). The use of marker terms to record significant structural information closely parallels the use of structured configurations of types in various sequent and natural deduction logical formulations. However, marker terms provide a more concise/readable representation of the proof's significant structural information.[2] In a linguistic context, a marker may be viewed as describing the linguistic structure of the object derived.

(1)
$$\frac{s \vdash \mathrm{A} \overset{\alpha}{\leftarrow} \mathrm{B} : a \qquad t \vdash \mathrm{B} : b}{(s \circ_\alpha t) \vdash \mathrm{A} : (a\ b)} \overset{\alpha}{\leftarrow}\mathrm{E} \qquad \frac{\begin{array}{c}[v \vdash \mathrm{B} : v] \\ (s \circ_\alpha v) \vdash \mathrm{A} : a\end{array}}{s \vdash \mathrm{A} \overset{\alpha}{\leftarrow} \mathrm{B} : \lambda v.a} \overset{\alpha}{\leftarrow}\mathrm{I}$$

(2)
$$\frac{t \vdash \mathrm{B} : b \qquad s \vdash \mathrm{B} \overset{\alpha}{\rightarrow} \mathrm{A} : a}{(t \circ_\alpha s) \vdash \mathrm{A} : (a\ b)} \overset{\alpha}{\rightarrow}\mathrm{E} \qquad \frac{\begin{array}{c}[v \vdash \mathrm{B} : v] \\ (v \circ_\alpha s) \vdash \mathrm{A} : a\end{array}}{s \vdash \mathrm{B} \overset{\alpha}{\rightarrow} \mathrm{A} : \lambda v.a} \overset{\alpha}{\leftarrow}\mathrm{I}$$

(3)
$$\frac{\begin{array}{c}[v \vdash \mathrm{B} : v], [w \vdash \mathrm{C} : w] \qquad t \vdash \mathrm{B} \circ_\alpha \mathrm{C} : b \\ s[(v \circ_\alpha w)] \vdash \mathrm{A} : a\end{array}}{s[t] \vdash \mathrm{A} : [b/(v \circ w)].a} \circ_\alpha\mathrm{E} \qquad \frac{s \vdash \mathrm{A} : a \qquad t \vdash \mathrm{B} : b}{(s \circ_\alpha t) \vdash \mathrm{A} \circ_\alpha \mathrm{B} : \langle a, b \rangle} \circ_\alpha\mathrm{I}$$

[1] We note particularly Hepple [5, 7] and Moortgat & Oehrle [10], who introduce multimodal frameworks of the similar character to the one that is described in this paper, although there are other proposals that are multimodal in the sense of including multiple groups of operators within a single system, e.g. Morrill [11], Morrill & Solias [12].

[2] Note that in (3), $s[(v \circ_\alpha w)]$ and $s[t]$ refer to marker terms that are identical except that where $(v \circ_\alpha w)$ appears as a subterm in one, t appears in the other.

Additional *structural rules*, which directly reflect axioms of the underlying semantics, act to modify the form of the marker and thereby play a crucial role in determining the properties of the derivability relation. For example, the associativity structural rule [a] in (4), which mirrors the associativity axiom $(x \circ_i (y \circ_i z)) = ((x \circ_i y) \circ_i z)$, is required to allow derivation of the 'composition' theorem $X \overset{i}{\leftarrow} Y$, $Y \overset{i}{\leftarrow} Z \Rightarrow X \overset{i}{\leftarrow} Z$, as illustrated in (5), and also the further example (6). The theorems of (5,6) mirror the theorems X/Y, Y/Z ⇒ X/Z and (X/Z)/Y, Y•Z ⇒ X of the associative Lambek calculus (**L**). Indeed, a system with only the single modality i plus the rules (1–4) constitutes a formulation of **L**. Further examples of structural rules are the rules of permutation and linkage shown in (7,8).

(4)
$$\dfrac{s[(x \circ_i (y \circ_i z))] \vdash A : a}{s[((x \circ_i y) \circ_i z)] \vdash A : a}\,[a]$$

(5)
$$\dfrac{x \vdash X \overset{i}{\leftarrow} Y : x \qquad \dfrac{y \vdash Y \overset{i}{\leftarrow} Z : y \qquad [z \vdash Z : z]}{(y \circ_i z) \vdash Y : (yz)}}{\dfrac{\dfrac{(x \circ_i (y \circ_i z)) \vdash X : (x(yz))}{((x \circ_i y) \circ_i z) \vdash X : (x(yz))}\,[a]}{(x \circ_i y) \vdash X : \lambda z.(x(yz))}}$$

(6)
$$\dfrac{\dfrac{x \vdash (X \overset{i}{\leftarrow} Z) \overset{i}{\leftarrow} Y : x \qquad [y \vdash Y : y]}{(x \circ_i y) \vdash X \overset{i}{\leftarrow} Z : (xy)} \qquad [z \vdash Z : z] \qquad w \vdash Y \circ_i Z : w}{\dfrac{\dfrac{((x \circ_i y) \circ_i z) \vdash X : ((xy)z)}{(x \circ_i (y \circ_i z)) \vdash X : ((xy)z)}\,[a]}{(x \circ_i w) \vdash X : [w/(y \circ z)].((xy)z)}}$$

(7)
$$\dfrac{s[(x \circ_i y)] \vdash A : a}{s[(x \circ_j y)] \vdash A : a}\,[i/j]$$

(8)
$$\dfrac{s[(x \circ_i y)] \vdash A : a}{s[(y \circ_i x)] \vdash A : a}\,[p]$$

The question remains of how the linear order (i.e. word order) consequences of derivations are determined. We cannot simply look at the linear order of assumptions written on the page, since not all modalities carry simple ordering import. For example, for a permutative modality $\circ_i$ (i.e. subject to (8)), we cannot take a marker $(x \circ_i y)$ as indicating that a word corresponding to the assumption labelled x precedes one for the assumption labelled y. Instead, we must identify a subset of modalities which do have simple linear order import, and take word/linear order to be indicated only by markers that are constructed using only those modalities.

2.2 Atomic Decomposition of Structural Relations

The multimodal approach favours a linguistic model with rich lexical encoding of syntactic information, i.e. not only word order and subcategorisation, but also (e.g.) default bracketting, head-dependent relations, etc. Where part of this information is not needed, it can be discarded in a move to a less informative mode of description, one that encodes only the information that is relevant to the treatment of the phenomenon at issue. Such an approach, however, threatens a bewildering proliferation of modes, each with its own associated axioms. Partly in response to this problem, this paper will explore the idea that the multiple modes are not simply individually stipulated, but instead arise via a set S of *structural atoms*, with each mode being indexed by some subset of S, i.e. each mode being realised via an operator $\circ_\alpha$, where $\alpha \subseteq S$. Under this approach, only the single linkage axiom (9) is required. Other axioms are specified schematically, i.e. requiring that certain atoms be present/absent. Since structural atoms can only be discarded, and given that linkage allows

movement to less informative descriptions, it follows that structural atoms should in general correspond to *restrictions* on structural freedom, i.e. so that as atoms are discarded, greater structural freedom, and hence less informative descriptions, result.

(9) $\quad x \circ_\alpha y \longrightarrow x \circ_\beta y \qquad (\beta \subseteq \alpha)$

Some notational conventions are required. The atom sets that index modes are written as atom 'strings', e.g. as in $\circ_{xy}$ for $\{x, y\}$. Obviously $\circ_{xy} = \circ_{yx}$. We will use $\alpha.\beta$ to indicate an atom set having subsets α and β *which are disjoint.* Hence, for example, equating $\circ_\alpha = \circ_{\beta.x}$ indicates that $\alpha = \beta \cup \{x\}$ and that $x \notin \beta$.

For illustration, let us consider a system having two structural atoms n and o, which correspond to the structural restrictions of *non-associativity* and *non-permutativity* ('ordered'), respectively. The system requires the axioms (10a,b), whose side conditions *exclude* the presence of the relevant atoms. This atom set gives rise to four modes, i.e. the modes indexed by the subsets: $\{n, o\}$, $\{n\}$, $\{o\}$ and $\emptyset$. Taken alone, these modes would correspond to the familiar systems **NL** (non-associative Lambek calculus), **NLP** (non-associative Lambek calculus with permutation), **L** (associative Lambek calculus), and **LP** (the Lambek-van Benthem calculus), respectively.

(10) a. $\quad (x \circ_\alpha (y \circ_\alpha z)) = ((x \circ_\alpha y) \circ_\alpha z) \qquad (n \notin \alpha)$

b. $\quad (x \circ_\alpha y) = (y \circ_\alpha x) \qquad (o \notin \alpha)$

A simple transitive verb *saw* might have the lexical type $(\mathrm{np} \overset{n\,o}{\rightarrow} \mathrm{s}) \overset{n\,o}{\leftarrow} \mathrm{np}$ (i.e. employing operators of the most restrictive mode *no*, where both order and bracketting are preserved) allowing the derivation (11) of *Kim saw Lee*. For this system, it is natural to accept either of the non-permutative modes *o* or *no* as indicating word order. Hence the final marker in (11) does indicate the intended word order *Kim saw Lee*.

(11)
$$\dfrac{kim \vdash \mathrm{np} : \mathbf{kim}' \qquad \dfrac{saw \vdash (\mathrm{np} \overset{n\,o}{\rightarrow} \mathrm{s}) \overset{n\,o}{\leftarrow} \mathrm{np} : \mathbf{saw}' \qquad lee \vdash \mathrm{np} : \mathbf{lee}'}{(saw \circ_{no} lee) \vdash \mathrm{np} \overset{n\,o}{\rightarrow} \mathrm{s} : (\mathbf{saw}'\ \mathbf{lee}')}}{(kim \circ_{no} (saw \circ_{no} lee)) \vdash \mathrm{s} : (\mathbf{saw}'\ \mathbf{lee}'\ \mathbf{kim}')}$$

Since bracketting is preserved in *no*, we cannot (e.g.) derive *kim saw* as $\mathrm{s} \overset{n\,o}{\leftarrow} \mathrm{np}$. But, since movement from mode *no* to mode *o* is possible under linkage, we *can* derive *kim saw* as $\mathrm{s} \overset{o}{\leftarrow} \mathrm{np}$, i.e. as follows:

(12)
$$\dfrac{\dfrac{\dfrac{\dfrac{kim \vdash \mathrm{np} : \mathbf{kim}' \qquad \dfrac{saw \vdash (\mathrm{np} \overset{n\,o}{\rightarrow} \mathrm{s}) \overset{n\,o}{\leftarrow} \mathrm{np} : \mathbf{saw}' \qquad v \vdash \mathrm{np} : v}{(saw \circ_{no} v) \vdash \mathrm{np} \overset{n\,o}{\rightarrow} \mathrm{s} : (\mathbf{saw}'\ v)}}{(kim \circ_{no} (saw \circ_{no} v)) \vdash \mathrm{s} : (\mathbf{saw}'\ v\ \mathbf{kim}')}}{(kim \circ_o (saw \circ_o v)) \vdash \mathrm{s} : (\mathbf{saw}'\ v\ \mathbf{kim}')}\,[\subseteq^*]}{((kim \circ_o saw) \circ_o v) \vdash \mathrm{s} : (\mathbf{saw}'\ v\ \mathbf{kim}')}\,[\mathrm{a}]}{(kim \circ_o saw) \vdash \mathrm{s} \overset{o}{\leftarrow} \mathrm{np} : \lambda v.(\mathbf{saw}'\ v\ \mathbf{kim}')}$$

Similarly, since order is preserved in the mode *o*, as it is in *no*, we cannot derive a type such as $\mathrm{s} \overset{o}{\leftarrow} \mathrm{np}$ for *kim saw today* (as in e.g. *... who kim saw _ today*), but, since neither bracketting nor order are preserved in mode $\emptyset$, we can derive a type $\mathrm{s} \leftarrow \mathrm{np}$, as in (13) (using undecorated $\leftarrow$ and $\circ$ to represent the operators of mode $\emptyset$).

(13)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{kim \vdash \text{np} : \mathbf{kim}' \quad \dfrac{saw \vdash (\text{np}\overset{n}{\rightarrow}\text{s})\overset{n}{\leftarrow}\text{np} : \mathbf{saw}' \quad v \vdash \text{np} : v}{(saw \circ_{no} v) \vdash \text{np}\overset{n}{\rightarrow}\text{s} : (\mathbf{saw}'\ v)} \quad today \vdash \text{s}\overset{n}{\rightarrow}\text{s} : \mathbf{today}'}{(kim \circ_{no} (saw \circ_{no} v)) \vdash \text{s} : (\mathbf{saw}'\ v\ \mathbf{kim}')}}{((kim \circ_{no} (saw \circ_{no} v)) \circ_{no} today) \vdash \text{s} : (\mathbf{today}'\ (\mathbf{saw}'\ v\ \mathbf{kim}'))}[\subseteq^*]}{((kim \circ (saw \circ v)) \circ today) \vdash \text{s} : (\mathbf{today}'\ (\mathbf{saw}'\ v\ \mathbf{kim}'))}[\text{a,p}^*]}{(((kim \circ saw) \circ today) \circ v) \vdash \text{s} : (\mathbf{today}'\ (\mathbf{saw}'\ v\ \mathbf{kim}'))}}{((kim \circ saw) \circ today) \vdash \text{s}\leftarrow\text{np} : \lambda v.(\mathbf{today}'\ (\mathbf{saw}'\ v\ \mathbf{kim}'))}$$

2.3 Eliminating Explicit Marker Terms

The labelling of formulae with both marker and semantic terms is not a necessary feature of this multimodal approach. The two terms can be eliminated in favour of a single lambda term that fully records the structure of the proof (i.e. which records not only instances of applications and abstractions but also their associated modalities). Such a lambda term makes explicit markers redundant, since the set of possible markers that could be associated with any proof that is labelled with this lambda term can be directly computed from the information that the term records. We shall develop a new formulation along these lines over the next three subsections. The ultimate value of this move, as we shall see, is to enable an approach in which lexical items may be associated with complex lexical string terms, constructed using the operators of the proof labelling algebra, to give what effectively amounts to lexical encoding of partial proof structure.

In the new formulation, labelled formulae take the form: $A : s$, with A a type, s a richly labelled lambda term. The natural deduction rules are now as in (14–16). Note that the formulation includes no structural rules, since there are no explicit marker terms on which they could operate.

(14)
$$\dfrac{\text{A}\overset{\alpha}{\leftarrow}\text{B} : a \quad \text{B} : b}{\text{A} : (a\ b)_{\overset{\leftarrow}{\alpha}}}\overset{\alpha}{\leftarrow}\text{E} \qquad \dfrac{[\text{B} : v]\ \vdots\ \text{A} : a}{\text{A}\overset{\alpha}{\leftarrow}\text{B} : \lambda_{\overset{\leftarrow}{\alpha}} v.a}\overset{\alpha}{\leftarrow}\text{I}$$

(15)
$$\dfrac{\text{B} : b \quad \text{B}\overset{\alpha}{\rightarrow}\text{A} : a}{\text{A} : (b\ a)_{\overset{\rightarrow}{\alpha}}}\overset{\alpha}{\rightarrow}\text{E} \qquad \dfrac{[\text{B} : v]\ \vdots\ \text{A} : a}{\text{B}\overset{\alpha}{\rightarrow}\text{A} : \lambda_{\overset{\rightarrow}{\alpha}} v.a}\overset{\alpha}{\rightarrow}\text{I}$$

(16)
$$\dfrac{[\text{B} : v],[\text{C} : w] \quad \text{B}\circ_\alpha\text{C} : b \quad \text{A} : a}{\text{A} : [b/(v \circ_\alpha w)].a}\circ_\alpha\text{E} \qquad \dfrac{\text{A} : a \quad \text{B} : b}{\text{A}\circ_\alpha\text{B} : \langle a, b\rangle_\alpha}\circ_\alpha\text{I}$$

The method for ensuring correct inferencing involves reconstructing for any proof the marker terms that could be associated with comparable proofs under the original formulation. The proof (17) illustrates key aspects of the use of markers in relation to the original formulation. Elimination inferences combine together the markers of preceding formulae. Structural rule uses modify the end-formula's marker. Commonly, there is more than one way that a given marker can be modified, i.e. there is a *set* of possible markers that could result from applying structural rules. The final introduction step modifies the preceding marker, provided that it satisfies certain form requirements.

(17)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{x \vdash \mathrm{X}\overset{\eta\,o}{\leftarrow}\mathrm{Y} : x \quad \dfrac{y \vdash \mathrm{Y}\overset{\eta\,o}{\leftarrow}\mathrm{Z} : y \quad [z \vdash \mathrm{Z} : z]}{(y \circ_{no} z) \vdash \mathrm{Y} : (yz)}}{(x \circ_{no} (y \circ_{no} z)) \vdash \mathrm{X} : (x(yz))}}{(x \circ_{o} (y \circ_{o} z)) \vdash \mathrm{X} : (x(yz))}[\subseteq^*]}{((x \circ_{o} y) \circ_{o} z) \vdash \mathrm{X} : ((xy)z)}[\mathrm{a}]}{(x \circ_{o} y) \vdash \mathrm{X}\overset{o}{\leftarrow}\mathrm{Z} : \lambda z.((xy)z)}$$

We will define a mapping Σ, which returns for any proof term of the new formulation, the set of markers that could be associated with corresponding proofs of the original formulation. The definition will include cases for application steps in proof terms that combine subterm marker sets, and cases for abstraction steps that modify subterm markers which satisfy certain form requirements, and so on. What would be missing if each case simply reflected a term construction step is the effects of structural rules. These effects can be allowed for by including marker rewriting steps in some of the defining cases. The following defining clauses are required (in which $p \rightsquigarrow q$ indicates that q derives from p under some sequence, possibly null, of structural rule rewrites):

$\Sigma.1$	$\Sigma(v)$	$= \{ v \mid v \in \mathtt{atomic} \}$
$\Sigma.2$	$\Sigma((a\ b)_{\overleftarrow{\alpha}})$	$= \{ z \mid \exists x, y.\ x \in \Sigma(a)\ \wedge\ y \in \Sigma(b)\ \wedge\ (x \circ_\alpha y)\ \rightsquigarrow\ z \}$
$\Sigma.3$	$\Sigma((a\ b)_{\overrightarrow{\alpha}})$	$= \{ z \mid \exists x, y.\ x \in \Sigma(a)\ \wedge\ y \in \Sigma(b)\ \wedge\ (x \circ_\alpha y)\ \rightsquigarrow\ z \}$
$\Sigma.4$	$\Sigma(\lambda_{\overleftarrow{\alpha}} v.a)$	$= \{ z \mid (z \circ_\alpha v) \in \Sigma(a) \}$
$\Sigma.5$	$\Sigma(\lambda_{\overrightarrow{\alpha}} v.a)$	$= \{ z \mid (v \circ_\alpha z) \in \Sigma(a) \}$
$\Sigma.6$	$\Sigma([b/(v \circ_\alpha w)].a)$	$= \{ z \mid x[(v \circ_\alpha w)] \in \Sigma(a)\ \wedge\ y \in \Sigma(b)\ \wedge\ x[y]\ \rightsquigarrow\ z \}$
$\Sigma.7$	$\Sigma(\langle a, b \rangle_\alpha)$	$= \{ z \mid x \in \Sigma(a)\ \wedge\ y \in \Sigma(b)\ \wedge\ (x \circ_\alpha y)\ \rightsquigarrow\ z \}$

Note the cases here which lack marker rewriting. For $\Sigma.1$, the marker is atomic and so cannot be affected by rewriting. For the abstraction cases $\Sigma.4$ and $\Sigma.5$, marker rewriting adds nothing (i.e. since any result w that could be derived by a rewrite $z\ \rightsquigarrow\ w$ will also result from a rewrite of the form $(v \circ_\alpha z)\ \rightsquigarrow\ (v \circ_\alpha w)$ at an earlier step plus subtraction of y at this step).

Consider how Σ allows us to distinguish correct and incorrect proofs. Given the correspondence between the defining cases of Σ and the steps of proof construction for the original formulation, the existence of any marker in $\Sigma(t)$ (where t is a proof term produced under the new formulation) shows that there is some sequence of logical and structural inferences that would allow us to derive the same result under the original formulation. If, on the other hand, the set $\Sigma(t)$ is empty, there is no such sequence of inferences under the original formulation, and so the proof at issue is not correct. In short then, a proof with proof term t is correct *iff* $\Sigma(t) \neq \emptyset$.

(18)

$$\dfrac{\dfrac{\mathrm{X}\overset{\eta\,o}{\leftarrow}\mathrm{Y} : x \quad \dfrac{\mathrm{Y}\overset{\eta\,o}{\leftarrow}\mathrm{Z} : y \quad [\mathrm{Z} : z]}{\mathrm{Y} : (yz)_{\overleftarrow{no}}}}{\mathrm{X} : (x(yz)_{\overleftarrow{no}})_{\overleftarrow{no}}}}{\mathrm{X}\overset{o}{\leftarrow}\mathrm{Z} : \lambda_{\overleftarrow{o}} z.(x(yz)_{\overleftarrow{no}})_{\overleftarrow{no}}}$$

(19)

$$\dfrac{\dfrac{\mathrm{X}\overset{\eta\,o}{\leftarrow}\mathrm{Y} : x \quad \dfrac{\mathrm{Y}\overset{\eta\,o}{\leftarrow}\mathrm{Z} : y \quad [\mathrm{Z} : z]}{\mathrm{Y} : (yz)_{\overleftarrow{no}}}}{\mathrm{X} : (x(yz)_{\overleftarrow{no}})_{\overleftarrow{no}}}}{\mathrm{X}\overset{\eta\,o}{\leftarrow}\mathrm{Z} : \lambda_{\overleftarrow{no}} z.(x(yz)_{\overleftarrow{no}})_{\overleftarrow{no}}}$$

For example, proof (18) is correct, and proof (19) incorrect (the latter since the combination requires (implicit) associativity, but the result indicates a non-associative mode). The

proof term of (18) has a subterm $(x(yz)_{\overleftarrow{no}})_{\overleftarrow{no}}$, whose marker set includes $(x \circ_{no} (y \circ_{no} z))$, plus its variants under rewriting from [$\subseteq$,a,p], including $((x \circ_o y) \circ_o z)$ and $((x \circ y) \circ_o z)$. Hence, the marker set of $\lambda_{\overleftarrow{o}} z.(x(yz)_{\overleftarrow{no}})_{\overleftarrow{no}}$ includes $(x \circ_o y)$ and $(x \circ y)$, showing that the same theorem is derivable under the original formulation (e.g. see proof (17) w.r.t. $(x \circ_o y)$). The proof term of the incorrect proof (21), however, has an empty marker set.

Since proofs no longer include explicit markers, we must accordingly modify the basis for identifying their word order consequences to instead require the presence within the proof term's marker set of a marker that is constructed using only appropriate (i.e. ordering) modalities.

2.4 Normalisation and Possible Proofs

We can define β-normalisation of the typed lambda terms used for proof labelling via the following contraction rules (notated $\rightsquigarrow_\phi$), which divide into two groups: beta cases (β) and commutative cases (c).

(20)

$$
\begin{array}{lcl}
((\lambda_{\overleftarrow{\alpha}} v.a)\ b)_{\overleftarrow{\alpha}} & \overset{\beta}{\rightsquigarrow}_\phi & a[b/v] \\
(b\ (\lambda_{\overrightarrow{\alpha}} v.a))_{\overrightarrow{\alpha}} & \overset{\beta}{\rightsquigarrow}_\phi & a[b/v] \\
[\langle b, c\rangle_\alpha / (v \circ_\alpha w)].a & \overset{\beta}{\rightsquigarrow}_\phi & a[b/v, c/w] \\
 & & \\
(([c/(v \circ_\alpha w)].a)\ b)_{\overleftarrow{\kappa}} & \overset{c}{\rightsquigarrow}_\phi & [c/(v \circ_\alpha w)].(a\ b)_{\overleftarrow{\kappa}} \\
(b\ ([c/(v \circ_\alpha w)].a))_{\overrightarrow{\kappa}} & \overset{c}{\rightsquigarrow}_\phi & [c/(v \circ_\alpha w)].(b\ a)_{\overrightarrow{\kappa}} \\
(([c/(v \circ_\alpha w)].b)\ a)_{\overrightarrow{\kappa}} & \overset{c}{\rightsquigarrow}_\phi & [c/(v \circ_\alpha w)].(b\ a)_{\overrightarrow{\kappa}} \\
(a\ ([c/(v \circ_\alpha w)].b))_{\overleftarrow{\kappa}} & \overset{c}{\rightsquigarrow}_\phi & [c/(v \circ_\alpha w)].(a\ b)_{\overleftarrow{\kappa}} \\
\langle([c/(x \circ_\kappa y)].a), b\rangle_\alpha & \overset{c}{\rightsquigarrow}_\phi & [c/(x \circ_\kappa y)].\ \langle a, b\rangle_\alpha \\
\langle a, ([c/(x \circ_\kappa y)].b)\rangle_\alpha & \overset{c}{\rightsquigarrow}_\phi & [c/(x \circ_\kappa y)].\ \langle a, b\rangle_\alpha \\
[([c/(v \circ_\kappa w)].b)/(x \circ_\alpha y)].a & \overset{c}{\rightsquigarrow}_\phi & [c/(v \circ_\kappa w)].([b/(x \circ_\alpha y)].a)
\end{array}
$$

The beta cases mirror unnecessary 'detours' in proofs where a connective is introduced and then immediately eliminated. The commutation cases relate to semantically non-distinct alternatives for ordering a [$\circ$E] instance before or after some other inference. Commuting contractions may be required to allow the identification and elimination of otherwise implicit beta redexes, e.g. as in the rewrite:

$$(([c/(x \circ y)].(\lambda_{\leftarrow} v.a))b)_{\leftarrow} \ \rightsquigarrow_\phi \ [c/(x \circ y)].((\lambda_{\leftarrow} v.a)b)_{\leftarrow} \ \rightsquigarrow_\phi \ [c/(x \circ y)].a[b/v]$$

Normalisation defines equivalence of terms, i.e. two terms are equivalent *iff* they have the same normal form. However, consideration of equivalence brings up an unusual characteristic of the approach as it is currently formalised, namely that it is possible to have terms that are equivalent, where one describes a correct proof and the other does not. For example, the terms (21a,b), corresponding to proofs (22,23) respectively, are equivalent (having (21c) as their normal form), but only proof (22) is correct. The proofs derive the theorem X$\overset{o}{\leftarrow}$Y $\Rightarrow$ X$\overset{o}{\leftarrow}$Y, differing in that the first proceeds via an intermediate type X$\overset{o}{\leftarrow}$Y (again), whereas the second has an intermediate type X$\leftarrow$Y. The second proof is incorrect because this intermediate type X$\leftarrow$Y does not preserve the mode information of the initial assumption, and hence the marker set for the subsequent elimination does not include

any elements of the form $(a \circ_o y)$ to license the final introduction step. In short, structural information that is discarded at one stage is lost for later inferences.

(21) a. $\lambda_{\overleftarrow{o}} y.((\lambda_{\overleftarrow{o}} v.(x\ v)_{\overleftarrow{o}})\ y)_{\overleftarrow{o}}$

b. $\lambda_{\overleftarrow{o}} y.((\lambda_{\leftarrow} v.(x\ v)_{\overleftarrow{o}})\ y)_{\leftarrow}$

c. $\lambda_{\overleftarrow{o}} y.(x\ y)_{\overleftarrow{o}}$

(22)
$$\dfrac{\dfrac{\dfrac{\dfrac{\mathrm{X}\overset{o}{\leftarrow}\mathrm{Y} : x \quad [\mathrm{Y} : v]}{\mathrm{X} : (x\ v)_{\overleftarrow{o}}} \quad [\mathrm{Y} : y]}{\mathrm{X}\overset{o}{\leftarrow}\mathrm{Y} : \lambda_{\overleftarrow{o}}\mathrm{v}.(x\ v)_{\overleftarrow{o}}}}{\mathrm{X} : ((\lambda_{\overleftarrow{o}} v.(x\ v)_{\overleftarrow{o}})\ y)_{\overleftarrow{o}}}}{\mathrm{X}\overset{o}{\leftarrow}\mathrm{Y} : \lambda_{\overleftarrow{o}}\mathrm{y}.((\lambda_{\overleftarrow{o}} v.(x\ v)_{\overleftarrow{o}})\ y)_{\overleftarrow{o}}}$$

(23)
$$\dfrac{\dfrac{\dfrac{\dfrac{\mathrm{X}\overset{o}{\leftarrow}\mathrm{Y} : x \quad [\mathrm{Y} : v]}{\mathrm{X} : (x\ v)_{\overleftarrow{o}}} \quad [\mathrm{Y} : y]}{\mathrm{X}\leftarrow\mathrm{Y} : \lambda_{\leftarrow}\mathrm{v}.(x\ v)_{\overleftarrow{o}}}}{\mathrm{X} : ((\lambda_{\leftarrow} v.(x\ v)_{\overleftarrow{o}})\ y)_{\leftarrow}}}{\mathrm{X}\overset{o}{\leftarrow}\mathrm{Y} : \lambda_{\leftarrow}\mathrm{y}.((\lambda_{\overleftarrow{o}} v.(x\ v)_{\overleftarrow{o}})\ y)_{\leftarrow}}$$

This paradoxical situation (i.e. having equivalent terms that differ in correctness) can be resolved by modifying the correctness criterion so that Σ is applied not to proof terms themselves, but rather to their normal forms. Since equivalent terms have the same normal form, they cannot then differ in correctness. The effect of adopting this modified correctness condition is to expand the set of possible proofs, e.g. allowing (23) as an additional proof of the same theorem as (24). Note that this move has no effect on the derivability relation. Firstly, any theorem now accepted was previously accepted. Thus, a theorem now accepted has a proof whose term has a normal form having a non-empty marker set, but that normal form term itself describes a proof of the same theorem that is accepted under both the original and modified versions of the correctness condition. Secondly, any theorem previously accepted is still now accepted. The proof of this point involves firstly showing for each contraction rule in (20) that applying the contraction rule to a term generates another term whose marker set includes the marker set of the input term. It follows, by simple induction, that any term having a non-empty marker set has a normal form that also has a non-empty marker set, and hence that any proof previously accepted is still now accepted.

2.5 Including lexical information in proofs

For the original formulation, the existence of separate marker and semantic terms with each formula provides a straightforward basis for the inclusion of lexical string and lexical semantic information in linguistic derivations, e.g. as in (11–13). How can such lexical information be included in proofs of the new formulation, where there are no explicit marker terms, and where the lambda term labels encode mode distinctions that seem inappropriate for natural language semantic purposes?

Lexical assumptions in proofs will be associated with a single term $\langle \textsc{string}, \textsc{sem} \rangle_{lex}$, which pairs together the lexical item's string and semantic terms, so that the proof term returned by any derivation includes both aspects of lexical information. Such a term, however, can be viewed differently depending on whether we are interested in its 'string' or semantic consequences. This idea of 'taking different views' is implemented via two reduction procedures. The first, 'string reduction' (notated $\rightsquigarrow_\phi$), reduces a proof term to the appropriate form for proof correctness testing using Σ, and requires the contraction rules for normalisation in (20) plus the additional 'lexical projection' rule in (24), which simply returns the string component of a lexical pair. The 'semantic reduction' procedure (notated $\rightsquigarrow_\sigma$), defined in (25), also requires a lexical projection rule, plus some 'delta' rules (δ) which

delete unwanted modality markings. We will also include in the definition the contraction rules (26) which effect beta normalisation, although this is not required by the account.

(24) $\langle a, b\rangle_{lex} \quad \overset{\pi}{\rightsquigarrow}_{\phi} \quad a$

(25)
$$\begin{array}{lll} \langle a, b\rangle_{lex} & \overset{\pi}{\rightsquigarrow}_{\sigma} & b \\ \lambda_{\underset{\alpha}{\leftarrow}} v.a & \overset{\delta}{\rightsquigarrow}_{\sigma} & \lambda v.a \\ \lambda_{\underset{\alpha}{\rightarrow}} v.a & \overset{\delta}{\rightsquigarrow}_{\sigma} & \lambda v.a \\ (a\ b)_{\underset{\alpha}{\leftarrow}} & \overset{\delta}{\rightsquigarrow}_{\sigma} & (a\ b) \\ (b\ a)_{\underset{\alpha}{\rightarrow}} & \overset{\delta}{\rightsquigarrow}_{\sigma} & (a\ b) \end{array}$$

(26)
$$\begin{array}{lll} ((\lambda v.a)\ b) & \overset{\beta}{\rightsquigarrow}_{\sigma} & a[b/v] \\ [\langle b, c\rangle\ /(v \circ w)].a & \overset{\beta}{\rightsquigarrow}_{\sigma} & a[b/v, c/w] \\ (([c/(v \circ w)].a)\ b) & \overset{c}{\rightsquigarrow}_{\sigma} & [c/(v \circ w)].(a\ b) \\ (a\ ([c/(v \circ w)].b)) & \overset{c}{\rightsquigarrow}_{\sigma} & [c/(v \circ w)].(a\ b) \\ [([c/(v \circ w)].b)/(x \circ y)].a & \overset{c}{\rightsquigarrow}_{\sigma} & [c/(v \circ w)].([b/(x \circ y)].a) \\ \langle([c/(x \circ y)].a), b\rangle & \overset{c}{\rightsquigarrow}_{\sigma} & [c/(x \circ y)].\ \langle a, b\rangle \\ \langle a, ([c/(x \circ y)].b)\rangle & \overset{c}{\rightsquigarrow}_{\sigma} & [c/(x \circ y)].\ \langle a, b\rangle \end{array}$$

Consider this approach in relation to proof (27). String reduction of the proof term yields $(kim\ (saw\ lee)_{\underset{no}{\leftarrow}})_{\underset{no}{\rightarrow}}$, whose marker set includes $(kim \circ_{no} (saw \circ_{no} lee))$, showing that the proof is correct and that the word order *Kim saw Lee* is indicated. Semantic reduction yields the undecorated term: (**saw′ lee′**) **kim′**).

(27)
$$\dfrac{\dfrac{\dfrac{\dfrac{\mathrm{np} : \langle kim, \mathbf{kim'}\rangle_{lex} \quad \dfrac{(\mathrm{np} \overset{n\,o}{\rightarrow} \mathrm{s}) \overset{n\,o}{\leftarrow} \mathrm{np} : \langle saw, \mathbf{saw'}\rangle_{lex} \quad [\mathrm{np} : x]}{\mathrm{np} \overset{n\,o}{\rightarrow} \mathrm{s} : (\langle saw, \mathbf{saw'}\rangle_{lex}\ x)_{\underset{no}{\leftarrow}}} \quad \mathrm{np} : \langle lee, \mathbf{lee'}\rangle_{lex}}{\mathrm{s} : (\langle kim, \mathbf{kim'}\rangle_{lex}, (\langle saw, \mathbf{saw'}\rangle_{lex}\ x)_{\underset{no}{\leftarrow}})_{\underset{no}{\rightarrow}}}}{\mathrm{s} \overset{o}{\leftarrow} \mathrm{np} : \lambda_{\underset{o}{\leftarrow}} \mathrm{x}.(\langle kim, \mathbf{kim'}\rangle_{lex}, (\langle saw, \mathbf{saw'}\rangle_{lex}\ x)_{\underset{no}{\leftarrow}})_{\underset{no}{\rightarrow}}}}{\mathrm{s} : ((\lambda_{\underset{o}{\leftarrow}} x.(\langle kim, \mathbf{kim'}\rangle_{lex}, (\langle saw, \mathbf{saw'}\rangle_{lex}\ x)_{\underset{no}{\leftarrow}})_{\underset{no}{\rightarrow}})\ \langle lee, \mathbf{lee'}\rangle_{lex})_{\underset{o}{\leftarrow}}}$$

Given that string reduction combines both lexical projection and beta normalisation into a single procedure, an interesting possibility arises. Namely that instead of providing a word with a simple atom for its lexical string component, we might instead provide a complex term that is built using the operators of the proof term algebra. String reduction could then act to fold this lexical term in with the proof term, so that the two together determine the correctness of a proof. Since any proof term encodes the structure of a natural deduction proof, such a move would amount to allowing lexical encoding of partial proof structure. We shall see some examples of this approach in the following subsections.

2.6 Linguistic Example 1: Quantification

The potential value of allowing lexical encoding of partial proof structure can be illustrated for quantification. A possible lexical assignment for *someone* is given in (28). Note the complex string term, which is one that would arise under a 'type raising' transformation such as np ⇒ s←(s←np). This type allows the derivation (29) of *John gave someone*

money. (Note that, due to space limitations, the proof has only lexical string terms in the place where lexical string/semantic pairs should appear.)

(28) $\mathrm{s}{\leftarrow}(\mathrm{s}{\leftarrow}\mathrm{np}) : \langle \lambda_{\leftarrow} v.(v\ someone)_{\leftarrow}, \mathbf{someone'} \rangle_{lex}$

(29)

$$\dfrac{\mathrm{s}{\leftarrow}(\mathrm{s}{\leftarrow}\mathrm{np}) : \lambda_{\leftarrow} v.(v\ someone)_{\leftarrow} \quad \dfrac{\mathrm{np} : john \quad \dfrac{((\mathrm{np}\overset{n\,s}{\rightarrow}\mathrm{s})\overset{p\,o}{\leftarrow}\mathrm{np})\overset{p\,o}{\leftarrow}\mathrm{np} : gave \quad [\mathrm{np} : v]}{(\mathrm{np}\overset{n\,s}{\rightarrow}\mathrm{s})\overset{p\,o}{\leftarrow}\mathrm{np} : (gave\ v)_{\overset{\leftarrow}{n\,o}}} \quad \mathrm{np} : money}{\mathrm{np}\overset{n\,s}{\rightarrow}\mathrm{s} : ((gave\ v)_{\overset{\leftarrow}{n\,o}}\ money)_{\overset{\leftarrow}{n\,o}}}}{\mathrm{s} : (john\ ((gave\ v)_{\overset{\leftarrow}{n\,o}}\ money)_{\overset{\leftarrow}{n\,o}})_{\overset{\rightarrow}{n\,o}}}}{\mathrm{s}{\leftarrow}\mathrm{np} : \lambda_{\leftarrow} v.(john\ ((gave\ v)_{\overset{\leftarrow}{n\,o}}\ money)_{\overset{\leftarrow}{n\,o}})_{\overset{\rightarrow}{n\,o}}}$$

$$\mathrm{s} : ((\lambda_{\leftarrow} v.(v\ someone)_{\leftarrow})\ \lambda_{\leftarrow} v.(john\ ((gave\ v)_{\overset{\leftarrow}{n\,o}}\ money)_{\overset{\leftarrow}{n\,o}})_{\overset{\rightarrow}{n\,o}})_{\leftarrow}$$

String and semantic reduction of the proof term yield the results (30a,b). The notation $\mapsto_{\Sigma}$ in (30a) is used to point out a significant element of the reduced term's marker set, one indicating the word order *John gave someone money* (shown to the right of $\Rightarrow$). Note how the quantifier's 'type raised' string term combines with the term for the rest of the sentence and applies it to the string atom *someone*, so that it is ultimately the verb that determines the quantifier's word order position.[3] Multiple quantifier cases are handled by this account as it stands, i.e. with alternative scopings being derived.

(30) a. $\rightsquigarrow_{\phi}\ (john\ ((gave\ someone)_{\overset{\leftarrow}{n\,o}}\ money)_{\overset{\leftarrow}{n\,o}})_{\overset{\rightarrow}{n\,o}}$
$\mapsto_{\Sigma}\ (john \circ_{no} ((gave \circ_{no} someone) \circ_{no} money))$
$\Rightarrow$ *John gave someone money*

b. $\rightsquigarrow_{\sigma}\ \mathbf{someone'}(\lambda v.(\mathbf{gave'}\ \mathbf{money'}\ v\ \mathbf{john'}))$

2.7 Linguistic Example 2: Extraction

Another case requiring a complex lexical string term is *wh*-movement. Adapting a standard approach within categorial grammar, the relative pronoun assignment (31) treats it as a function that combines with a 'sentence missing NP', the latter being categorised simply as $\mathrm{s}{\leftarrow}\mathrm{np}$, allowing that the NP can be 'missing' from anywhere within the embedded clause. The associated string term is one that applies the proof term of the 'sentence missing NP' to ϵ, the 'null string' atom, and thereby fixes the site of the missing NP as being empty. An example derivation is given in (32) (again with only string terms included), and the results of applying the reduction procedures to its proof term are shown in (33). The use of the null string atom plays a key role here in allowing that the full structure of the embedded clause is recovered under normalisation. Without it, the normalised string term would still involve abstraction over the position of the missing NP, and this term would not yield a marker that was sufficient to determine word order. The lexical string term of (28), unlike the quantifier's, does not encode to a correct proof. In its use of the null string atom, it effectively involves an instance of deletion. We might view the term as importing proof structure that, by lexical stipulation, goes beyond the limits of the syntactic logic to a broader logic including rules such deletion, and so on.

(31) $\mathrm{rel}\overset{p\,o}{\leftarrow}(\mathrm{s}{\leftarrow}\mathrm{np}) : \left\langle \lambda_{\overset{\leftarrow}{n\,o}} s.(which\ (s\ \epsilon)_{\leftarrow})_{\overset{\leftarrow}{n\,o}},\ \mathbf{which'} \right\rangle_{lex}$

[3] See Morrill & Solias [12], Hepple [6], and Oehrle [13] for related proposals.

(32)

(which)	(john)	(bought)		(today)
$\text{rel}\overset{?\,o}{\leftarrow}(\text{s}\leftarrow\text{np}) : \lambda_{no}^{\leftarrow} s.(which\ (s\ \epsilon)\leftarrow)_{no}^{\leftarrow}$	$\text{np} : john$	$(\text{np}\overset{n\,s}{\rightarrow}\text{s})\overset{?\,o}{\leftarrow}\text{np} : bought$	$[\text{np} : v]$	$\text{s}\overset{n\,s}{\rightarrow}\text{s} : today$

$$\frac{\dfrac{\dfrac{\dfrac{\dfrac{\text{np}\overset{n\,s}{\rightarrow}\text{s} : (bought\ v)_{no}^{\leftarrow}}{\text{s} : (john\ (bought\ v)_{no}^{\leftarrow})_{no}^{\rightarrow}}}{\text{s} : ((john\ (bought\ v)_{no}^{\leftarrow})_{no}^{\rightarrow}\ today)_{no}^{\rightarrow}}}{\text{s}\leftarrow\text{np} : \lambda\leftarrow v.((john\ (bought\ v)_{no}^{\leftarrow})_{no}^{\rightarrow}\ today)_{no}^{\rightarrow}}}{\text{rel} : ((\lambda_{no}^{\leftarrow} s.(which\ (s\ \epsilon)\leftarrow)_{no}^{\leftarrow})\ (\lambda\leftarrow v.((john\ (bought\ v)_{no}^{\leftarrow})_{no}^{\rightarrow}\ today)_{no}^{\rightarrow}))_{no}^{\leftarrow}}$$

(33) a. $\rightsquigarrow_{\phi}\ ((\lambda_{no}^{\leftarrow} s.(which\ (s\ \epsilon)\leftarrow)_{no}^{\leftarrow})\ (\lambda\leftarrow v.((john\ (bought\ v)_{no}^{\leftarrow})_{no}^{\rightarrow}\ today)_{no}^{\rightarrow}))_{no}^{\leftarrow}$
$\mapsto_{\Sigma}\ (which \circ_{no} ((john \circ_{no} (bought \circ_{no} \epsilon)) \circ_{no} today))$
$\Rightarrow$ *which john bought today*

b. $\rightsquigarrow_{\sigma}\ (\mathbf{which}'\ (\lambda v.(\mathbf{today}'\ (\mathbf{bought}'\ v\ \mathbf{john}'))))$

3 Head Promotion within a Multimodal Grammar

According to a theory of grammatical relations that has been used within Montague Grammar, grammatical relations map to specific argument places, e.g. a subject is the last argument sought by a verb, a direct object the next-to-last argument, and so on (Dowty [3]). Lexical argument order provides an ordering over grammatical relations, one that corresponds to the traditional notion of grammatical hierarchy or relative obliqueness, i.e. so that any earlier argument of the verb is more oblique than any later argument, the subject being the least oblique argument of all. Combining this theory with a purely concatenative grammatical framework leads to a prediction that (e.g.) the most oblique argument of a verb should always appear adjacent to it, which is contradicted by examples such as *Lee gave Kim vodka* where the second object *vodka* is most oblique. Such problems are avoided within Montague Grammar by use of the non-concatenative string operation *wrap* [1] (e.g. with the TVP string *gave vodka* being 'wrapped' around the object string *kim* in deriving the VP *gave Kim vodka*).

Jacobson [8], working with a hybrid categorial/phrase structure framework, avoids these problems in a different way, by adopting the idea that English has an 'underlying' order where verbs *follow* their direct objects, and hence are adjacent to any second object, if present. Observed word order results via bounded movement of the verb to VP initial position, giving e.g.: $[_{\text{VP}}\ [_{\text{V}}\ \text{gave}]_i\ [_{\text{NP}}\ \text{kim}]\ [_{\text{TVP}}\ t_i\ [_{\text{NP}}\ \text{vodka}]\ ]\ ]$

In this section, a treatment of English word order is described which is inspired by Jacobson's proposal, and which allows the above treatment of grammatical hierarchy to be adopted within a multimodal grammar. The following structural atoms are required:

$\succ$	:	left element is head
$\prec$	:	right element is head
o	:	non-permuting ('ordered')
c	:	'command preserved'
i	:	internal argument
u	:	non-abstract ('utterable')

Following Moortgat & Morrill [9], the account employs modes that distinguish between heads and dependents, as indicated by the presence of either the structural atom $\succ$ or $\prec$. In $(x \circ_{\succ\alpha} y)$, x is head and y its dependent, whereas y is head in $(x \circ_{\prec\alpha} y)$. To make

head-dependent relations more explicit, we will allow that a headed mode such as $\circ_{\succ\alpha}$ can be written as $\succ_\alpha$, e.g. as in $(x \succ_\alpha y)$. (This notation is intended to be reminiscent of the 'arrow structures' of dependency grammar, where heads 'point' at their dependents.) The atom c marks arguments for which command information is preserved, as will be explained in the next section. As before, atom o indicates non-permutativity, and atom i identifies the internal (i.e. non-subject) verb arguments. The account makes crucial use of a distinction between abstract and non-abstract modes: only the latter, which include the atom u, are 'phonetically interpretable', i.e. allow for word order determination. More specifically, it is stipulated that word order determination should involve only the non-abstract mode $\circ_{ou}$.

The essence of the account is that internal arguments of English verbs lack the u atom, but, in the space of possible marker terms, this feature may be gained as a side effect of a restructuring in which the head element is (left) 'promoted' over the argument. That a head may undergo such promotion is lexically marked by the presence of the operator $^\triangle$. For example, the ditransitive verb *gave* has a lexical assignment as follows, whose argument order accords with obliqueness, and whose lexical string is marked with $^\triangle$.

(34) $\quad (\mathrm{np} \overset{\prec_{oci}}{\rightarrow} (\mathrm{np} \overset{\prec_{ocu}}{\rightarrow} \mathrm{s})) \overset{\succ_{oci}}{\leftarrow} \mathrm{np} : \left\langle {}^\triangle gave,\ \mathbf{gave'} \right\rangle_{lex}$

(35)

$$\dfrac{\dfrac{\dfrac{}{\mathrm{np} : lee} \quad \dfrac{}{\mathrm{np} : kim} \quad \dfrac{\dfrac{}{(\mathrm{np} \overset{\prec_{oci}}{\rightarrow} (\mathrm{np} \overset{\prec_{ocu}}{\rightarrow} \mathrm{s})) \overset{\succ_{oci}}{\leftarrow} \mathrm{np} : {}^\triangle gave} \quad \dfrac{}{\mathrm{np} : vodka}}{\mathrm{np} \overset{\prec_{oci}}{\rightarrow} (\mathrm{np} \overset{\prec_{ocu}}{\rightarrow} \mathrm{s}) : ({}^\triangle gave\ vodka)_{\overset{\leftarrow}{\succ_{oci}}}}}{\mathrm{np} \overset{\prec_{ocu}}{\rightarrow} \mathrm{s} : (kim\ ({}^\triangle gave\ vodka)_{\overset{\leftarrow}{\succ_{oci}}})_{\overset{\rightarrow}{\prec_{oci}}}}}{\mathrm{s} : (lee\ (kim\ ({}^\triangle gave\ vodka)_{\overset{\leftarrow}{\succ_{oci}}})_{\overset{\rightarrow}{\prec_{oci}}})_{\overset{\rightarrow}{\prec_{ocu}}}}$$

The derivation (35) yields the results shown in (36a,b) under reduction. To allow for the promotion operator, the definition of Σ must be extended with the extra clause (37), which not only provides a corresponding operator in the marker domain, but also introduces a null string atom that serves as a placeholder for the verb atom as it promotes. The string term (36a) yields the 'maximal' marker (36c). This marker contains 'abstract' modes (i.e. lacking u) and so cannot itself provide a direct basis for determining word order.

(36) a. $\quad \rightsquigarrow_\phi \ (lee\ (kim\ ({}^\triangle gave\ vodka)_{\overset{\leftarrow}{\succ_{oci}}})_{\overset{\rightarrow}{\prec_{oci}}})_{\overset{\rightarrow}{\prec_{ocu}}}$

b. $\quad \rightsquigarrow_\sigma \ (\mathbf{gave'\ vodka'\ kim'\ lee'})$

c. $\quad \mapsto_\Sigma \ (lee \prec_{ocu} (kim \prec_{oci} ((^\triangle gave \succ_{ou} \epsilon) \succ_{oci} vodka)))$

(37) $\quad \Sigma.8 \qquad \Sigma(^\triangle a) \ = \ \{ (^\triangle y \succ_{ou} \epsilon) \mid x \in \Sigma(a) \ \wedge \ x \rightsquigarrow y \}$

Additional marker axioms that involve the promotion operator are shown in (38). Axiom (38a) allows a promotion operator to be discarded. Axioms (38b,c) allow a head that is marked with the promotion operator to be promoted over its internal arguments, those arguments being made non-abstract as a side effect. Such a pattern of restructuring is shown in (39), where the final marker is one that indicates the word order *Lee gave Kim vodka.*

(38) a. $\quad ^\triangle x \longrightarrow x$

b. $\quad (^\triangle x \succ_\alpha y) \succ_{\beta.i} z \ = \ ^\triangle x \succ_\alpha (y \succ_{\beta.iu} z)$

c. $\quad z \prec_{\beta.i} (^\triangle x \succ_\alpha y) \ = \ ^\triangle x \succ_\alpha (z \prec_{\beta.iu} y)$

(39) $(lee \prec_{ocu} (kim \prec_{oci} ((^{\triangle} gave \succ_{ou} \epsilon) \succ_{oci} vodka)))$
$\rightsquigarrow (lee \prec_{ocu} (kim \prec_{oci} (^{\triangle} gave \succ_{ou} (\epsilon \succ_{ociu} vodka))))$
$\rightsquigarrow (lee \prec_{ocu} (^{\triangle} gave \succ_{ou} (kim \prec_{ociu} (\epsilon \succ_{ociu} vodka))))$
$\rightsquigarrow (lee \prec_{ocu} (gave \succ_{ou} (kim \prec_{ociu} (\epsilon \succ_{ociu} vodka))))$
$\rightsquigarrow (lee \circ_{ou} gave \circ_{ou} kim \circ_{ou} \epsilon \circ_{ou} vodka)$

Note how this account does not strictly involve 'movement'. Rather, amongst the multiple descriptions of a derived object that the multimodal approach allows, those which are relevant to word order display a restructuring analogous to verb movement. Note also that the 'movement' allowed by this method is inherently bounded, i.e. would not allow a head to move out of its own domain. This approach has potential for application to other phenomena involving bounded head movement, e.g. of the finite verb in V2 languages.

4 Command Relations in Binding

One approach to handling reflexives in Categorial Grammar has involved making them a function over verb types, with a semantics that equates two verb argument positions [15]. Challenges for such an approach include incorporating the limitations on binding that are handled via c-command in PSG approaches whilst still allowing the permissible cases, other than by simply assigning a list of lexical categories that separately encode each possibility. Within Montague Grammar, Bach & Partee [2] handle command constraints via a condition *F-command* on argument structure (whereby any argument F-commands any 'earlier' (and hence more oblique) arguments of the same functor and their subconstituents). A related *o-command* condition has been used within Head-driven Phrase Structure Grammar, which recreates F-command in terms of subcategorisation lists [14].

As in the previous section, we use an approach where relative obliqueness is encoded via lexical argument order. This encoding provides the basis for handling command relations. A key advantage of the multimodal approach is that it will allow us to exploit a level of description where information relevant to the treatment of binding is encoded, but where other information has been discarded, with the benefit of avoiding the need to assign multiple lexical entries where this would otherwise have been required. The level of description we require is one where headedness still counts but linear order does not, so the axiom (40a) is included, which undermines the latter whilst maintaining the former.

(40) a. $x \succ_\alpha y \;=\; y \prec_\alpha x \qquad (o \notin \alpha)$

b. $x \prec_\alpha (y \succ_\beta z) \;=\; (x \prec_\alpha y) \succ_\beta z \qquad (c \notin \alpha \;\vee\; c \notin \beta)$

The axiom (40b) is a form of associativity — one which reverses the relative hierarchical position of two dependents of the same head. If freely operative, this axiom would undermine the relative obliqueness information initially encoded by lexical argument order, but the axiom is instead restricted to apply only provided one or both dependencies lack the c ('command preserved') atom. Hence, two dependencies that retain this atom must maintain the relative hierarchical position between them that was lexically given. The flexibility of this level of description is such that a marker can restructure to the form $((x \succ_c y) \succ_c z)$ *iff* z corresponds to an argument that F-commands y. Furthermore, since y and z must be dependents of the same head, a form of locality is imposed, one which is comparable to that used in the binding account of Pollard & Sag [14]. The reflexive assignment (41), used in the derivation (42) of *John showed Mary himself*, exploits this level of description.

(41) $(\mathrm{s}^{\succ^c}\!\!\leftarrow \mathrm{np}) \leftarrow (\mathrm{s}^{\succ^c}\!\!\leftarrow \mathrm{np}^{\succ^c}\!\!\leftarrow \mathrm{np}) : \langle \lambda f.(f\ himself)_{\leftarrow}, \lambda f \lambda x.(f x x) \rangle_{lex}$

(42)

(himself)		(mary)	(showed)		(john)
$(s\overset{\succ c}{\leftarrow}np)\leftarrow(s\overset{\succ c}{\leftarrow}np\overset{\succ c}{\leftarrow}np)$: $\lambda f.(f\ himself)\leftarrow$	$[np:z]$	$np:mary$	$(np\overset{\prec_{ocu}}{\rightarrow}(np\overset{\prec_{oci}}{\rightarrow}s))\overset{\succ_{oci}}{\leftarrow}np$: $^{\triangle}showed$	$[np:y]$	$np:john$

$$\frac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{np\overset{\prec_{ocu}}{\rightarrow}(np\overset{\prec_{oci}}{\rightarrow}s):(^{\triangle}showed\ y)_{\succ_{oci}\leftarrow}}}{np\overset{\prec_{ocu}}{\rightarrow}s:(mary\ (^{\triangle}showed\ y)_{\succ_{oci}\leftarrow})_{\prec_{oci}\rightarrow}}}{s:(z\ (mary\ (^{\triangle}showed\ y)_{\succ_{oci}\leftarrow})_{\prec_{oci}\rightarrow})_{\prec_{ocu}\rightarrow}}}{s\overset{\succ c}{\leftarrow}np:\lambda{\leftarrow}z.(z\ (mary\ (^{\triangle}showed\ y)_{\succ_{oci}\leftarrow})_{\prec_{oci}\rightarrow})_{\prec_{ocu}\rightarrow}}}{s\overset{\succ c}{\leftarrow}np\overset{\succ c}{\leftarrow}np:\lambda{\leftarrow}y\lambda{\leftarrow}z.(z\ (mary\ (^{\triangle}showed\ y)_{\succ_{oci}\leftarrow})_{\prec_{oci}\rightarrow})_{\prec_{ocu}\rightarrow}}}{s\overset{\succ c}{\leftarrow}np:((\lambda f.(f\ himself)\leftarrow)\ (\lambda{\leftarrow}y\lambda{\leftarrow}z.(z\ (mary\ (^{\triangle}showed\ y)_{\succ_{oci}\leftarrow})_{\prec_{oci}\rightarrow})_{\prec_{ocu}\rightarrow}))\leftarrow}}{s:(((\lambda f.(f\ himself)\leftarrow)\ (\lambda{\leftarrow}y\lambda{\leftarrow}z.(z\ (mary\ (^{\triangle}showed\ y)_{\succ_{oci}\leftarrow})_{\prec_{oci}\rightarrow})_{\prec_{ocu}\rightarrow}))\leftarrow john)_{\succ_c\leftarrow}}$$

(43) a. $\rightsquigarrow_{\phi}\ (john\ (mary\ (^{\triangle}showed\ himself)_{\succ_{oci}\leftarrow})_{\prec_{oci}\rightarrow})_{\prec_{ocu}\rightarrow}$

$\mapsto_{\Sigma}\ (john\ \prec_{ocu}\ (showed\ \succ_{ou}\ (mary\ \prec_{ociu}\ (\epsilon\ \succ_{ociu}\ himself))))$

b. $\rightsquigarrow_{\sigma}$ **(showed′ john′ mary′ john′)**

Note that the proof includes additional assumptions corresponding to the binder and reflexive positions (labelled with variables z and y respectively). These assumptions combine with the verb, plus a further NP, to derive a sentence category subproof with the string term shown in (44a), which yields, amongst others, the marker in (44b). Marker (44b) restructures to the form $((a\ \succ_c\ y)\ \succ_c\ z)$, as shown in (44c). Consequently, the assumptions labelled z and y can be discharged in introduction inferences to yield a result $s\overset{\succ c}{\leftarrow}np\overset{\succ c}{\leftarrow}np$, which forms the argument of the reflexive type. Normalisation ensures that the word order positions of both the reflexive and the binder are as determined by the verb.

(44) a. $(z\ (mary\ (^{\triangle}showed\ y)_{\succ_{oci}\leftarrow})_{\prec_{oci}\rightarrow})_{\prec_{ocu}\rightarrow}$

b. $\mapsto_{\Sigma}\ \ (z\ \prec_{ocu}\ (mary\ \prec_{oci}\ (^{\triangle}showed\ \succ_{oci}\ y)))$

c. $\rightsquigarrow\ \ (z\ \prec_c\ (mary\ \prec\ (^{\triangle}showed\ \succ_c\ y)))$

$\rightsquigarrow\ \ (z\ \prec_c\ ((mary\ \prec\ ^{\triangle}showed)\ \succ_c\ y))$

$\rightsquigarrow\ \ (((mary\ \prec\ ^{\triangle}showed)\ \succ_c\ y)\ \succ_c\ z)$

This approach readily extends to allow for non-local command relations. Rewriting a marker $(x \succ (y \succ z))\ \rightsquigarrow\ ((x \succ y) \succ z)$ effectively amounts to moving the 'embedded' dependent z upwards to act as a dependent at a higher level. Axiom (45a) allows a restricted form of such restructuring, provided all dependencies are marked as 'command preserving', and ensuring that the upwardly moved dependency is marked with $\bar{c}$ (indicating '*non-local* command preservation') rather than c (for '*local* command preservation'). The local restructuring axiom must be modified to take account of both c and $\bar{c}$, as in (45b). In this system, a marker can restructure to the form $((x \succ_{\bar{c}} y) \succ_c z)$ *iff* argument z *non-locally* F-commands y, i.e. y originates from *within* an argument that is locally F-commanded by z. Hence, the lexical assignment (46) can be used for non-locally bound pronoun uses, e.g. as in *Every boy$_i$ thinks Mary likes him$_i$*.

(45) a. $x \succ_{\alpha.c} (y \succ_{\delta.j} z)\ =\ (x \succ_{\alpha.c} y) \succ_{\delta.\bar{c}} z \quad (j \in \{c, \bar{c}\})$

b. $x \prec_{\alpha} (y \succ_{\beta} z)\ =\ (x \prec_{\alpha} y) \succ_{\beta} z\ \ (c, \bar{c} \notin \alpha\ \vee\ c, \bar{c} \notin \beta)$

(46) $(s\overset{\succ c}{\leftarrow}np)\leftarrow(s\overset{\succ c}{\leftarrow}np\overset{\succ \bar{c}}{\leftarrow}np):\langle\lambda f.(f\ him)\leftarrow, \lambda f\lambda x.(f x x)\rangle_{lex}$

References

[1] Bach, E. 1981. 'Discontinuous Constituents in Generalized Categorial Grammars.' *NELS*, **11**, pp1–12.

[2] Bach, E. and Partee, B.H. 1980. 'Anaphora and semantic structure.' In K.J. Kreiman, and A.E. Ojeda (Eds), *Papers from the Parasession on Pronouns and Anaphora*, Chicago Linguistic Society.

[3] Dowty, D. 1982. 'Grammatical relations and Montague grammar.' In P. Jacobson and G.K. Pullum (Eds), *The Nature of Syntactic Representation*, D. Reidel, Dordrecht.

[4] Gabbay, D.M. 1994. *Labelled deductive systems. Part I: Foundations.* Oxford University Press (to appear).

[5] Hepple, M. 1993. 'A general framework for hybrid substructural categorial logics.' Ms, IRCS, UPenn. Available as IRCS Report 94-14.

[6] Hepple, M. 1994. 'Labelled deduction and discontinuous constituency'. In M. Abrusci, C. Casadio and M. Moortgat (eds). *Linear Logic and Lambek Calculus. Proceedings 1st Rome Workshop.* OTS Working Papers.

[7] Hepple, M. 1995. 'Mixing Modes of Linguistic Description in Categorial Grammar.' *Proceedings EACL–7.*

[8] Jacobson, P. 1987. 'Phrase Structure, Grammatical Relations and Discontinuous Constituents.' In Huck, G.J. and Ojeda, A.E. (Eds), *Syntax and Semantics*, **20**: *Discontinuous Constituency*, Academic Press, New York.

[9] Moortgat, M. & Morrill, G. 1991. 'Heads and Phrases: Type Calculus for Dependency and Constituency.' To appear: *Journal of Language, Logic and Information.*

[10] Moortgat, M. & Oehrle, R. 1994. 'Adjacency, dependency and order'. *Proceedings of Ninth Amsterdam Colloquium.*

[11] Morrill, G. 1995. 'Clausal Proofs and Discontinuity', *Bulletin of the Interest Group in Pure and Applied Logics*, Vol. 3, No. 2,3.

[12] Morrill, G. & Solias, M.T. 1993. 'Tuples, Discontinuity, and Gapping in Categorial Grammar.' *Proc. of EACL–6*, Utrecht.

[13] Oehrle, R. 1995. 'Some 3-Dimensional Systems of Labelled Deduction', *Bulletin of the Interest Group in Pure and Applied Logics*, Vol. 3, No. 2,3.

[14] Pollard, C. & Sag, I. 1992. 'Anaphors in English and the Scope of Binding Theory'. *Linguistic Inquiry* **23**, 261-303.

[15] Szabolcsi, A. 1987. 'Bound variables in syntax (are there any?)' In Groenendijk, J., Stokhof, M. and Veltman, F. (Eds), *Sixth Amsterdam Colloquium*, Institute for Language, Logic and Information, University of Amsterdam.