Meta-Categorial Grammar*

Glyn Morrill

0. *Introduction*

In this paper we present a theory of syntax called Meta-Categorial Grammar in which Categorial Grammar is augmented with metarules. We define the theory and then specify a grammar for English constructed within the framework. Expressions generated by the sample grammar include the following:

(1) [I gave _ and Mary sent _] A BOOK TO EACH STUDENT

(2) THE MODERN DANCING John said that he likes _!

(3) He LENT [_ John a book _ and _ Mary a paper _] ABOUT SUBJACENCY

(4) NUMEROUS STATUES _ [_ were destroyed OF THE OLD PRESIDENT and _ were erected OF THE NEW ONE]

(5) This is a woman ABOUT WHOM an argument _ _ started WHICH WENT ON ALL NIGHT

(6) This is a town WHICH he BOUGHT [_ a ticket to _ not wanting to visit _ and _ a ticket from _ not wanting to leave _]

'_'s mark the positions ('gaps') in which the capitalised subexpressions ('fillers') appear in the canonical counterparts to non-canonical expressions. In (1) both complements of the prepositional ditransitive verbs have been Right Node Raised out of the coordinate structure. Example (2) exhibits the potentially unbounded extraction of fronted constituents. In (3) there is simultaneous Left Node Raising of the ditransitive verb and Right Node Raising of the noun modifier from the coordinate structure, and in (4) right extraposed subject modifiers form conjuncts with the verb phrases which they have been extracted past. In (5) two subject modifiers have been extracted: one leftwards and one rightwards, and in (6) the transitive verb has been Left Node Raised from the coordinate structure and simultaneously there is parasitic extraction from the complements and adverbial phrases which

*I would like to express my gratitude to Bob Carpenter and Einar Jowsey for their assistance in the development of the ideas here presented; without their insight and expertise the work could not have proceeded the way it did. I would also like to extend my appreciation to all those others who have assisted me in this work; they include Elisabet Engdahl, Dick Hudson, Ewan Klein, Charles McCarty, Mark Steedman, and Max Volino. The errors in this paper are of course attributable only to me.

comprise each conjunct.

These examples are all instances of the general problem addressed in this paper, non-canonicality: the phenomenon exhibited in extraction and coordination wherein semantically cohesive expressions are discontiguous. In clauses exhibiting extraction, modifiers appear clause-initially or clause-finally rather than in the post-head positions which they occupy canonically. In coordinate structures conjuncts may not be canonical constituents. We assume that conjuncts are meaningful expressions so that an account of non-canonicality must explain how such non-canonical expressions are assigned meanings. Meta-Categorial Grammar (MCG) is a monostratal theory, that is a theory positing a single level of syntactic analysis, providing an account of non-canonicality. A meta-categorial grammar defines a hierarchy of binary context-free grammars, referred to as the zero-order or base grammar, the first-order grammar, the second-order grammar, and so on. These grammars generate languages exhibiting increasingly severe departures from canonicality. Because a meta-categorial grammar defines not a single language, but a hierarchy of languages, each member of which subsumes its predecessors, in MCG expressions are characterised not just as grammatical or ungrammatical, but grammatical expressions are characterised according to the order of the grammar by which they are generated. Such gradation seems consistent with the fact that acceptability is a matter of degree, not something absolute. Within MCG the base grammar, which generates canonical expressions, is a Categorial Grammar (Ajdukiewicz 1935). Higher-order grammars are generated from the base grammar by metarules which are the analogues of the transformations of Transformational Grammar and the metarules of Generalised Phrase Structure Grammar (GPSG; see Gazdar, Klein, Pullum and Sag 1985). Amongst the phenomena characterised are right node raising, topicalisation, relativisation, pied piping, 'non-constituent' coordination, heavy noun phrase shift, right extraposition, and parasitic extraction. We define the theory in section 1, and in section 2 we specify a grammar for English and take a look at the fragment it defines. In section 3 there is a discussion of the theory which includes a comparison of MCG metarules with transformations and GPSG metarules.

1. The Theory

We take as our point of departure the following principle:

(7) The Combinatory Principle
The meaning of an expression is a λI-function of the meanings of its immediate subexpressions.

By a **λI-function** we mean a function definable in the pure λI-calculus (Barendregt 1981). **Pure** means that the only operations available are application and abstaction: there are no constants. The significance of this is that meaning is never introduced syncategorematically: the semantics of combinations can only ever 'evaluate' meanings specified lexically. The λI-calculus is the λ-calculus less vacuous abstraction. In terms of combinators (Curry and Feys 1958), the λI-functions are those definable in terms of **I**, **C**, **B**, and **S**, but crucially not **K** which corresponds to vacuous abstraction. The significance of the absence of such abstraction is that a subexpression always contributes to the meaning of an expression containing it. The grammars developed in Ades and Steedman (1982), Steedman (1985, 1987, this volume) and Dowty (1985) can be viewed as based on the Combinatory Principle. Any such grammars are monostratal and compositional (Partee 1984); indeed the Combinatory Principle is our formulation of the Principle of Compositionality.

Insofar as it is possible we wish to remain uncommitted as to precisely what meanings are functions over, though what we have in mind are relational structures something like Lexical-Functional Grammar's f-structures (Kaplan and Bresnan 1982), Kamp's Discourse Representation Structures (Kamp 1981), or Webber's Level-1 representations (Webber 1979), in which some scope and anaphora may be unresolved, so that 'meaning' may underdetermine 'interpretation'. In order to abstract across all theories of meaning which would be consistent with our claims we will represent meanings by logical constants and logical formulas. Of course such formulas are not themselves meanings, but rather denote meanings. They do not have any theoretical status, they are only there to enable us to present a syntactic theory resting on general assumptions about meaning, without committing ourselves on details of meaning which are irrelevant to the theory.

A set Δ of basic types of meaning defines a set Types(Δ) as follows:

(8) $\quad \text{Types}(\Delta) = \Delta \cup \{\langle a, b\rangle : a, b \in \text{Types}(\Delta)\}$

$\langle a, b\rangle$ is the type of functions mapping from functions of type a into functions of type b. If our basic types include S and NP, the types of meanings of sentences and noun phrases, the set of types will include S, NP, $\langle NP, S\rangle$ and $\langle NP, \langle NP, S\rangle\rangle$. A set Δ of basic types also defines a set Cats(Δ) of categories and a function TYP_{Δ} mapping from categories to types (| is either / or \):

(9) a. $\text{Cats}(\Delta) = \Delta \cup \{X|Y: X \text{ and } Y \in \text{Cats}(\Delta)\}$

b. $TYP_{\Delta}(X) = X \text{ if } X \in \Delta$
$TYP_{\Delta}(X|Y) = \langle TYP_{\Delta}(Y), TYP_{\Delta}(X)\rangle$

Thus if our basic types include S and NP, the set of categories will include S\NP and

(S\NP)/NP, and $TYP_{\Delta}((S\backslash NP)/NP)$ = <NP, <NP, S>>.

In MCG all combinatory rules take two inputs, and hence all derivation paths are binary. A combinatory rule thus has four attributes: the λI-function that is the semantics of the rule, the categories of its left- and right-hand daughters, and the category of its mother. We will adopt the convention that the semantics of a rule applies to the meanings of its input subexpressions in left-to-right order to give the meaning of the mother expression. A **combinatory rule** is then defined to be a quadruple (ϕ, C_1, C_2, C_0) where C_1, C_2 and C_0 are categories and ϕ is a λI-function function which applied to a function of type $TYP_{\Delta}(C_1)$ and then to a function of type $TYP_{\Delta}(C_2)$ yields a function of type $TYP_{\Delta}(C_0)$. A set Δ of basic types and a set Γ of categories which is a subset of Cats(Δ), defines a set Rules(Δ, Γ) of possible combinatory rules as follows:

(10) Rules(Δ, Γ) = {(ϕ, C_1, C_2, C_0): C_0, C_1 and $C_2 \in \Gamma$ and ϕ is a λI-function which applied to a function of type $TYP_{\Delta}(C_1)$ and then to a function of type $TYP_{\Delta}(C_2)$ yields a function of type $TYP_{\Delta}(C_0)$}

Now we can define a **combinatory grammar** to be a triple (Δ, Γ, Π) where Δ is a set of basic types, Γ is a set of categories which is a subset of Cats(Δ), and Π is a set of combinatory rules which is a subset of Rules(Δ, Γ). It will be clear that a combinatory rule is just a binary context-free rewrite rule with a certain kind of semantics, and that a combinatory grammar is a context-free grammar. We define a **derivation path** to be a quadruple (T, L, C, ϕ) where T is a binary tree, L is an *n*-tuple of categories which label the *n* leaves of T, C is a category labelling the root of T, and ϕ is the semantics of the derivation path. The semantics of the derivation path is the function which applied in left-to-right order to the meanings of expressions of the categories of the leaves yields a meaning for the expression of category the root formed by that derivation path. Note that a derivation path contains no lexical items and is distinct from a **terminated derivation** in which the leaves are labelled with terminal symbols, i.e. words. The set Paths(Δ, Γ, Π) of derivation paths projected by a combinatory grammar (Δ, Γ, Π) is as shown in (11). We encode trees linearly by using asterisks for leaves and square brackets to indicate structure. In logical formulas throughout, application is indicated by juxtaposition, and is left-associative.

(11) $\text{Paths}(\Delta, \Gamma, \Pi) = \{(*, (C), C, \lambda x[x]): c \in \Gamma\} \cup$

$\{([T_1T_2], (C_{1,1}, C_{1,2}, ..., C_{1,n}, C_{2,1}, C_{2,2}, ..., C_{2,m}), C_0,$

$\lambda x_{1,1}\lambda x_{1,2}...\lambda x_{1,n}\lambda x_{2,1}\lambda x_{2,2}...\lambda x_{2,m}[\psi(\phi_1 x_{1,1}x_{1,2}...x_{1,n})(\phi_2 x_{2,1}x_{2,2}...x_{2,m})])$:

$(\psi, C_{1,0}, C_{2,0}, C_0) \in \Pi$, and

$(T_1, (C_{1,1}, C_{1,2}, ..., C_{1,n}), C_{1,0}, \phi_1) \in \text{Paths}(\Delta, \Gamma, \Pi)$ and

$(T_2, (C_{2,1}, C_{2,2}, ..., C_{2,m}), C_{2,0}, \phi_2) \in \text{Paths}(\Delta, \Gamma, \Pi)\}$

The first part of the definition in (11) says that the set of derivation paths includes trees consisting of single nodes. The leaf and root of such trees, which happen to be the same node, are labelled with the same category and the semantics of these derivation paths is the identity function. There are such elementary derivation paths for all of the categories in the grammar. The second part of the definition shows how rules build derivation paths out of subderivation paths. By way of illustration, suppose we have a combinatory grammar (Δ, Γ, Π) where Δ (the set of basic types) is {S, NP}, Γ (a set of categories which is a subset of Cats(Δ)) is {S, NP, S\NP, (S\NP)/NP}, and Π (a set of rules which is a subset of Rules(Δ, Γ)) is {(λxλy[xy], (S\NP)/NP, NP, S\NP), (λzλw[wz], NP, S\NP, S)}. Then Paths(Δ, Γ, Π) will include the elementary derivation paths (*, ((S\NP)/NP), (S\NP)/NP, λx[x]) and (*, (NP), NP, λx[x]). Building from these derivation paths using the rule (λxλy[xy], (S\NP)/NP, NP, S\NP) gives us the derivation path ([**], ((S\NP)/NP, NP), S\NP, $\lambda x_{1,1}\lambda x_{2,1}[\lambda x\lambda y[xy](\lambda i[i]x_{1,1})(\lambda j[j]x_{2,1})]$), that is ([**], ((S\NP)/NP, NP), S\NP, $\lambda x_{1,1}\lambda x_{2,1}[x_{1,1}x_{2,1}]$). Where this derivation path is the right-hand subderivation path, and (*, (NP), NP, λx[x]) is the left-hand one, the rule (λzλw[wz], NP, S\NP, S) builds the derivation path ([*[**]], (NP, (S\NP)/NP, NP), S, $\lambda x_{1,1}\lambda x_{2,1}\lambda x_{2,2}[\lambda z\lambda w[wz](\lambda i[i]x_{1,1})(\lambda x\lambda y[xy]x_{2,1}x_{2,2})]$, that is ([*[**]], (NP, (S\NP)/NP, NP), S, $\lambda x_{1,1}\lambda x_{2,1}\lambda x_{2,2}[x_{2,1}x_{2,2}x_{1,1}]$).

Derivation paths show how larger expressions can be formed from smaller ones. The basic expressions forming the starting point from which a combinatory grammar defines a language are provided by a **lexicon**:

(12) a. A lexical entry is a triple (W, C, M) where W is a word, C is a category and M is a function of type $\text{TYP}_\Delta(C)$.
b. A lexicon is a set of lexical entries.

Where the notions of an **expression** and a **language** are formalised as in (13), a set Φ of derivation paths and a lexicon Ξ define the language L(Φ, Ξ) as shown in (14).

(13) a. An expression is a triple (U, C, M) where U is an n-tuple of words, C is a category, and M is a function of type $TYP_{\Delta}(C)$.
b. A language is a set of expressions.

(14) $$L(\Phi, \Xi) = \{((W_1, W_2, ..., W_n), C_0, \phi M_1 M_2 ... M_n): (T, (C_1, C_2, ..., C_n), C_0, \phi) \in \Phi, \text{ and for } 1 \leq i \leq n, (W_i, C_i, M_i) \in \Xi\}$$

Informally, whenever the categories labelling the leaves of a derivation path match up with the lexical categories of words, those words can be strung together to form an expression whose meaning is given by applying the semantics of the derivation path to the meanings of the words in left-to-right order.

In combinatory grammars the semantics of a rule can be any λI-function. A subset of these grammars will adhere to the following principle:

(15) The Categorial Principle
The meaning of an expression is obtained by applying the meaning of one subexpression to the meaning of the other.

In such grammars, the semantics of a rule is either the function λxλy[xy] which applies the meaning of the first expression to that of the second, or λxλy[yx] which applies the meaning of the second expression to that of the first. We adopt a directional-slash convention (cf. Lambek 1961; Bach 1983) whereby an expression which applies forwards has a category A/B, and one which applies backwards has category A\B. In combinatory grammars respecting the Categorial Principle and adhering to this convention, all combinatory information is encoded in categories. Thus a **categorial grammar** is just a pair (Δ, Γ) where Δ is a set of basic types and Γ is a set of categories which is a subset of Cats(Δ). The combinatory grammar CG(Δ, Γ) so defined is given by:

(16) $$CG(\Delta, \Gamma) = (\Delta, \Gamma, \{(\lambda x \lambda y[xy], C_0/C_2, C_2, C_0): C_0/C_2 \in \Gamma\} \cup \{(\lambda x \lambda y[yx], C_1, C_0\backslash C_1, C_0): C_0\backslash C_1 \in \Gamma\})$$

By way of example, CG({S, NP}, {S, NP, S\NP, (S\NP)/NP}) is equal to ({S, NP}, {S, NP, S\NP, (S\NP)/NP}, {(λxλy[yx], NP, S\NP, S), (λxλy[xy], (S\NP)/NP, NP, S\NP)}).

Under MCG canonical expressions respect the Categorial Principle, non-canonical expressions respect the Combinatory Principle, and the operations underlying departures from canonicality are metarules which map from existing rules into new rules. A metacategorial grammar defines an infinite hierarchy of combinatory grammars, each member of which is a superset of its predecessors. The interpretation of a metarule is that if the rules in the *i*th-order combinatory grammar project a certain derivation path, then the *i+1*th-order grammar contains a certain rule. A **metarule** is defined to be a quadruple (*T*, *j*, *k*, |)

where

T is an *n*(= 3 or 4)-leaf binary tree, *j* and *k* are integers such that $1 \leq j < k \leq n$, and | is / or \. The rule output by a metarule will have as its daughter categories the categories labelling the *j*th and *k*th leaves of the derivation path that is its input, and as its mother category $C_0|C_r$ where C_0 was the category labelling the root of the input and C_r was the category on the remaining leaf or leaves of the input; the node(s) labelled C_r will be said to have been raised. It will be seen that the inputs to metarules are not arbitrarily large derivation paths, but only ones with 3 or 4 leaves. The analogue of this locality in Extended Standard Theory is subjacency. We cannot have less than 3 leaves in the input because the categories labelling two leaves are daughter categories in the output and there would be no node to raise. The 4-leaf cases are those where the raising of one node is 'parasitic' on the raising of another (see Engdahl 1983). We assume that there is never parasitic node raising of more than two nodes in a single metarule, hence the absence of metarules with more than 4 leaves (this does not mean that multiple parasitic gaps cannot arise through successive applications of a single metarule).

A **meta-categorial grammar** is defined to be a quadruple (Δ, Γ, Π, Σ) where Δ is a set of basic types, Γ, the set of base categories, is a subset of Cats(Δ), Π, the set of declared rules, is a subset of Rules(Δ, Γ), and Σ is a set of metarules. A meta-categorial grammar (Δ, Γ, Π, Σ) defines a hierarchy of combinatory grammars $MCG_0(\Delta, \Gamma, \Pi, \Sigma)$, $MCG_1(\Delta, \Gamma, \Pi, \Sigma)$, $MCG_2(\Delta, \Gamma, \Pi, \Sigma)$, ... as shown in (17); y^m indicates m (possibly = 0) occurances of y; note that when the membership condition in the middle clause is satisfied, n is instantiated to the number of leaves in the input tree.

(17) $MCG_i(\Delta, \Gamma, \Pi, \Sigma) = (\Delta, \Gamma_i, \Pi_i \cup \Pi'_i)$ where

$$\Pi_0 = \{\},$$

$$\Pi_{i+1} = \Pi \cup \{(\lambda a\lambda b\lambda c[\chi c^{j-1}ac^{k-j-1}bc^{n-k}], C_j, C_k, C_0|C_r):$$

$$(T, j, k, |) \in \Sigma, \text{ and}$$

$$(T, (C_r^{j-1}, C_j, C_r^{k-j-1}, C_k, C_r^{n-k}), C_0, \chi) \in \text{Paths}(MCG_i(\Delta, \Gamma, \Pi, \Sigma))\}$$

$$\Gamma_0 = \Gamma,$$

$$\Gamma_{i+1} = \Gamma_i \cup \{C: (\phi, C_1, C_2, C) \in \Pi_{i+1}\}$$

$$CG(\Delta, \Gamma_i) = (\Delta, \Gamma_i, \Pi'_i)$$

For i greater than zero, Π_i is the union of the set of combinatory rules that is the third parameter of the meta-categorial grammar (Δ, Γ, Π, Σ), with the set of rules obtained by applying the metarules to the derivation paths projected by the *i-1*th-order grammar; Γ_i is the union of the set of categories in the *i*th-order grammar with the set of mother categories in Π_i. For all i, Π'_i is the set of categorial combinatory rules defined by the categorial grammar (Δ, Γ_i). The *i*th-order grammar is (Δ, Γ_i, $\Pi_i \cup \Pi'_i$). By way of illustration of the application of metarules, applying the metarule ([*[**]], 1, 2, /) to the derivation path ([*[**]], (NP, (S\NP)/NP, NP), S, λxλyλz[yzx]) yields the combinatory rule (λaλbλc[λxλyλz[yzx]$c^{1-1}ac^{2-1-1}bc^{3-2}$], NP, (S\NP)/NP, S/NP) which simplifies to (λaλbλc[bca], NP, (S\NP)/NP, S/NP).

We have defined metarules as applying to derivation paths because this enables us to specify the semantics of any instance of any metarule, and to identify the class of possible metarules. However because the set of derivation paths projected by a set of rules is implicit in the set of rules, it is possible to view metarules as mapping directly from rules to rules. This is how we shall be viewing them in the next section.

2. *A Meta-Categorial Grammar for English*

We begin by defining a categorial grammar for canonical English. A categorial grammar it will be recalled is a pair (Δ, Γ) where Δ is a set of basic types and Γ is a subset of Cats(Δ). Δ and Γ will be the first two parameters of our meta-categorial grammar. The set Δ of basic types is {N, NP, S and S'}, the types of the meanings of nouns, noun phrases, sentences and complementized sentences. The set Γ of base categories is shown in figure 1. For typographical ease VP is used to abbreviate S\NP; it is not a basic type. Some illustrative lexical assignments are shown in figure 2. In this section a combinatory rule (ϕ, C_1, C_2, C_0) will be notated:

(18) $\phi >> C_1 + C_2 => C_0$

For categorial rules of forward application, that is those where the semantics is the application of the meaning of the first subexpression to that of the second, the semantics will be **f** = λxλy[xy], and for rules of backward application where the meaning of the second subexpression is applied to that of the first, it will be **b** = λxλy[yx]. Note how as usual we are writing combinators using boldface. The rules in the categorial combinatory grammar CG(Δ, Γ) we have just defined will include the following:

(19) a. **f** >> NP/N + N => NP
b. **f** >> (S\NP)/NP + NP => S\NP
c. **b** >> NP + S\NP => S

We borrow Steedman's notation for derivations, and we label derivation steps with combinator expressions denoting their semantics. For example the derivation of the canonical expression *I said that John liked the modern dancing* is written as follows:

```
(20)  I   said      that John  liked     the  modern dancing
      --  --------  ---- ----  --------- ---- ------ -------
      NP (S\NP)/S'  S'/S  NP  (S\NP)/NP NP/N  N/N     N
                                              ----------f
                                                    N
                                         ------------f
                                                 NP
                                 ----------------f
                                         S\NP
                            -------------b
                                   S
                     ------------f
                            S'
              ---------------f
                   S\NP
      ------------b
           S
```

Example (21) shows the derivation of *I gave a book to John*; the prepositional ditransitive verb is assumed to be type-lifted over the obligatory adverbial.

```
(21)  I        gave             a     book    to        John
      --  -------------------- ----  ----  ---------- ----
      NP  ((S\NP)/(VP\VP))/NP  NP/N   N    (VP\VP)/NP  NP
                               -------f    --------------f
                                  NP           VP\VP
          --------------------------f
              (S\NP)/(VP\VP)
              -------------------------------------f
                               S\NP
      ---------------------------b
                  S
```

If W' is the meaning of the word W, the meaning of (21) will be bI'(f(fgave'(fa'book'))(fto'John')), which is equal to gave'(a'book')(to'John')I'.

Our first metarule, which we call Right Node Raising, is ([*[**]], 1, 2, /). The metarule says that if we have rules projecting a 3-leaf right branching derivation path with root V and leaves X, Y and Z in left-to-right order, then we can have a rule combining X and Y to form V/Z. For the purpose of elucidation metarules and their instances will be stated using a 'proof-theoretic' notation due to Einar Jowsey which shows them mapping directly from rules into rules. Notated this way Right Node Raising is:

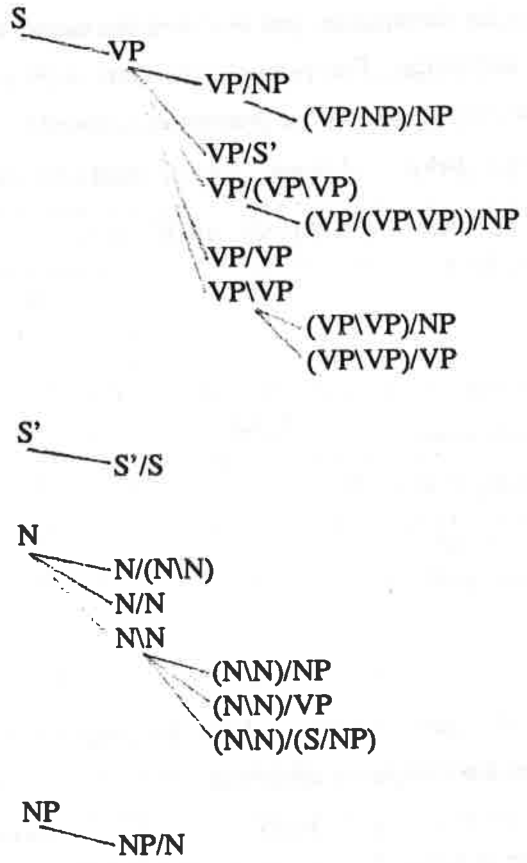

Figure 1: Base categories

VP	arrived, destroyed, erected, exists, started
VP/NP	are, bought, consulting, informing, implements, is, like, likes, read, reference
(VP/NP)/NP	gave, sent, lent
VP/S'	said
(VP/(VP\VP))/NP	gave, sent, lent, put
VP/VP	not, to, were, will
VP\VP	tonight, yesterday
(VP\VP)/NP	to, by
(VP\VP)/VP	without
S'/S	that
N	book, changes, dancing, executive, men, one, papers, president, student, subjacency, town, women
N/(N\N)	argument, brothers, picture, sisters
N/N	modern, new, old
(N\N)/NP	about, by, from, to
(N\N)/VP	who
(N\N)/(S/NP)	whom, which
NP	Beethoven, Bloomfield, Fred, he, I, John, Mary, these, this, you
NP/N	a, each, five, most, the, several, six, two

Figure 2: Lexical assignments

(22) ϕ >> Y + Z => W
ψ >> X + W => V

R$\phi\psi$ >> X + Y => V/Z
where **R**$\phi\psi = \lambda x\lambda y\lambda z[\psi x(\phi yz)]$ and Z $\in$ MOVEABLE

The value of the combinator **R** arises as follows. Recall first that the semantics of the output rule of a metarule $(T, j, k, |)$ is $\lambda a\lambda b\lambda c[\chi c^{j-1}ac^{k-j-1}bc^{n-k}]$ where n is the number of leaves in T and χ is the semantics of the input derivation path. In the case of Right Node Raising $n = 3$, $j = 1$ and $k = 2$ so that the semantics of the output in terms of the semantics χ of the input is

(23) $\lambda a\lambda b\lambda c[\chi acb]$

Now where the right-branching input derivation path is projected by two rules fitting the schema above the horizontal line in (22), the semantics of the derivation path, that is the function which applied in left-to-right order to the meanings of expressions dominated by the leaves yields the meaning of the expression dominated by the root, is $\lambda x\lambda y\lambda z[\psi x(\phi yz)]$.

Substituting this for χ in (23) gives $\lambda a\lambda b\lambda c[\lambda x\lambda y\lambda z[\psi x(\phi yz)]abc] = \lambda a\lambda b\lambda c[\psi a(\phi bc)]$.

In (22) a constraint has been specified on the node being raised. In general we shall be supplying metarules with such constraints. The interpretation of these metarules is not interestingly different from that of pure metarules and the formalisation of the framework with the constraints is left until the appendix. The constraint on Right Node Raising is that the category of the node raised be a member of MOVEABLE:

(24) MOVEABLE = {N\N, VP\VP, NP, S'}

The same constraint applies to the Middle Node Raising metarule that we introduce later.

Our first instance of Right Node Raising is one generating a construction which is already known by that name. In (25) we give the metarule instance, then an example of the construction, and then a derivation illustrating the non-canonical constituency apparently demanded by that construction. It will be clear from the derivation how the example could be generated via a naive coordination schema (26), coordination reduction operation (27), or schematic lexical assignment (28). All the coordination examples considered in this paper could be generated by any of these techniques. Our assumption is that the coordination operation requires its conjuncts to be generable as meaningful subexpressions, but we choose to remain agnostic as to the coordination operation itself.

```
(25) a. f >> (S\NP)/NP + NP => S\NP
        b >> NP + S\NP => S
        ------------------------------
        Rfb >> NP + (S\NP)/NP => S/NP

     b. [I liked _ and Mary adored _] THE MODERN DANCING

     c. I      liked      the modern dancing
        -- ------------- -------------------
        NP  (S\NP)/NP              NP
        -------------Rfb
           S/NP
           ------------------------f
                      S
```

(26) X + and + X => X

(27) [(A) B1 (C)], [(A) B2 (C)] --> [(A) B1 and B2 (C)]

(28) and := (X\X)/X

Since **Rfb** = $\lambda x\lambda y\lambda z[bx(fyz)]$ and **f** applies its first argument to its second while b applies its second to its first, **Rfb** is equal to $\lambda x\lambda y\lambda z[yzx]$. Thus the meaning of *I liked* in (25c) is λz[liked'zI'] and the meaning of the expression generated overall is the same as that of the expression when it is generated canonically. This illustrates how metarules, like their classical transformation analogues, are meaning preserving.

The Right Node Raising rule in (25a) generates the constituent with the gap in the topicalised sentence

(29) THE MODERN DANCING I liked _!

We use the following rule schema to introduce topics[1].

(30) Topicalisation
t >> X + S/X => S
where **t** = λxλy[yx] and X ∈ MOVEABLE

Thus (29) receives the analysis in (31); since the semantics of topicalisation, **t**, is just application, the meaning of the topicalised expression is the same as that of the canonical form.

```
(31)    The modern dancing  I    liked
        ------------------  --  ----------
                NP          NP  (S\NP)/NP
                            -------------Rfb
                                  S/NP
                ----------------------t
                             S
```

We could introduce topics via a metarule ([*[**]], 1, 2, \) which says that if there is a 3-leaf right branching derivation path, then there is a rule combining expressions of the categories labelling the first two leaves into expressions of a category which applies leftwards to expressions of the category labelling the third leaf to form expressions of the category labelling the root. Our principle reason for not doing this is that to appropriately constrain the application of such a metarule we would need to specify that the root is S while in general it appears appropriate to allow metarules to apply irrespective of the category of the root.

We assign subject relative pronouns the category (N\N)/(S\NP) and object relative pronouns the category (N\N)/(S/NP). The rule **Rfb** >> NP + (S\NP)/NP => S/NP immediately enables us to generate minimal cases of object relativization such as

```
(32)    (This is the dancing)   which        I     liked
                            -------------   --   ----------
                            (N\N)/(S/NP)    NP   (S\NP)/NP
                                            -------------Rfb
                                                   S/NP
                            -----------------------f
                                      N\N
```

[1]Meta-categorial grammars as we have defined them do not contain schematic rules. However the schematic rules we use have a finite number of instances and so can be encoded in the framework as it has been defined.

Right Node Raising of a subcategorized adverbial phrase is licenced in the same way as Right Node Raising of a direct object:

```
(33) a. f >> (S\NP)/(VP\VP) + VP\VP => S\NP
        b >> NP + S\NP => S
        ------------------------------------------
        Rfb >> NP + (S\NP)/(VP\VP) => S/(VP\VP)

     b. [I gave a book _ and Mary gave a paper _] TO EACH
                                                  STUDENT

     c. I    gave a book       to each student
        -    -------------     ---------------
        NP   (S\NP)/(VP\VP)        VP\VP
        -----------------Rfb
              S/(VP\VP)
            ------------------------f
                     S
```

Again topicalisation is derivative on the Right Node Raising:

```
(34)    To the student  he  gave a book
        --------------  --  --------------
           VP/VP        NP  (S\NP)/(VP\VP)
                        ------------------Rfb
                            S/(VP\VP)
           ------------------------t
                     S
```

In general sentences with 'gaps' are analysed as S/A where A is the category of the missing constituent. An expression of category S/A can of course apply forwards to expressions of category A to form sentences so our account predicts that constituents which can left-extract can also appear in sentence-final position.

To generate pied piping we assume the schematic rule

(35) Pied Piping
p >> X/NP + (N\N)/(S/NP) => (N\N)/(S|X)
where p = $\lambda x \lambda y \lambda z[y(\lambda w[z(xw)])]$ and X $\in$ {N\N, VP\VP, NP}

where as before | is either / or \. An example of pied piping is

```
(36)    (This is the student)   to          whom          I gave a book
                                ----------  ------------  -------------
                                (VP\VP)/NP  (N\N)/(S/NP)  S/(VP\VP)
                                ------------------------p
                                   (N\N)/(S/(VP\VP))
                                   --------------------------------f
                                                 N\N
```

One difference between GPSG metarules and MCG metarules is that while the former do not recurse, the latter are allowed to apply to derivation paths projected by rules which were themselves generated by metarules. Allowing the metarules to recurse in this way gives rise to multiple departures from canonicality such as the double Right Node Raising in (37).

```
(37) a. f >> ((S\NP)/(VP\VP))/NP + NP => (S\NP)/(VP\VP)
        Rfb >> NP + (S\NP)/(VP\VP) => S/(VP\VP)
        ------------------------------------------------------------
        Rf(Rfb) >> NP + ((S\NP)/(VP\VP))/NP => (S/(VP\VP))/NP

     b. [I gave _ _ and Mary sent _ _] A BOOK TO EACH STUDENT

     c. I         gave                 a book   to each student
        -- ------------------          ------   ---------------
        NP ((S\NP)/(VP\VP))/NP         NP       VP\VP
        -----------------------Rf(Rfb)
              (S/(VP\VP))/NP
              --------------------------f
                       S/(VP\VP)
                       ----------------------------------f
                                  S
```

Semantically, adnominals - relative clauses and adnominal prepositional phrases - modify nouns. However in examples such as (38b) and (39b) they appear to combine with coordinated noun phrases. The instances of Right Node Raising in (38a) and (39a) show how determiner-noun sequences can be analysed as noun phrases type-lifted over adnominals when these adnominals are subcategorized and when they are not respectively. We assume that *brother* and *sister* take of-prepositional phrase complements but that relative clauses are not complements.

```
(38) a. f >> N/(N\N) + N\N => N
        f >> NP/N + N => NP
        -------------------------------
        Rff >> NP/N + N/(N\N) => NP/(N\N)

     b. [the brothers _ and the sisters _] of John

(39) a. b >> N + N\N => N
        f >> NP/N + N => NP
        ---------------------------
        Rbf >> NP/N + N => NP/(N\N)

     b. [the men _ and the women _] WHO LIKE BEETHOVEN
```

A recursive application enables us to simultaneously Right Node Raise a subcategorized adnominal phrase and a non-subcategorized one:

```
(40) a. f >> N/(N\N) + N\N => N
        Rbf >> NP/N + N => NP/(N\N)
        -------------------------------------------------
        Rf(Rbf) >> NP/N + N/(N\N) => (NP/(N\N))/(N\N)

     b. [the brothers _ and the sisters _] OF JOHN WHO ARRIVED
                                                   YESTERDAY
```

As well as Right Node Raising an adnominal prepositional phrase, just the object of such a prepositional phrase can be Right Node Raised:

```
(41) a. f >> (N\N)/NP + NP => N\N
        f >> NP/(N\N) + N\N => NP
        ---------------------------------------
        Rff >> NP/(N\N) + (N\N)/NP => NP/NP

     b. [five tickets to _ and six tickets from _] A TOWN NEAR
                                                     SALTZBERG

     c. five tickets      to      a town near Saltzberg
        ------------  --------  -----------------------
         NP/(N\N)     (N\N)/NP            NP
         -------------------Rff
                  NP/NP
                  --------------------------f
                             NP
```

Being able to generate a determiner-noun-preposition sequence as a meaningful expression of category NP/NP means that we can capture subject pied piping such as that in (42).

```
(42)    a picture of       whom        exists
        ------------  --------------  ------
          NP/NP       (N\N)/(S/NP)    S\NP
          -------------------------p
               (N\N)/(S\NP)
               --------------------------f
                         N\N
```

Also, once we can Right Node Raise a preposition's object out of a noun phrase, the following instance of **Rff** enables the object to be fronted:

```
(43) a. f >> NP/NP + NP => NP
        f >> VP/NP + NP => VP
        -----------------------------
        Rff >> VP/NP + NP/NP => VP/NP

     b. (This is the town)   which        I   bought       a ticket to
                            ------------ --  ----------   -----------
                            (N\N)/(S/NP) NP  (S\NP)/NP      NP/NP
                                             ------------------Rff
                                                   (S\NP)/NP
                                         ------------------Rfb
                                                    S/NP
                            --------------------------f
                                         N\N
```

Any instance of **Rff** is an instance of Steedman's Forward Composition operation. Examples like the next two constitute strong evidence that generalised categorial grammars must achieve the effect of such an operation. For example in (44b) it seems that the meaning of the future auxiliary verb *will* must be a function over the meanings of bare infinitival verb phrases, and that the meaning of the transitive verb *reference* must be a function over the meanings of object noun phrases. Then the meaning of the conjunct *will reference* can only be the composition of these functions.

```
(44) a. f >> VP/NP + NP => VP
        f >> VP/VP + VP => VP
        -----------------------------
        Rff >> VP/VP + VP/NP => VP/NP

     b. I [read _ and will reference _] SEVERAL PAPERS BY
                                        BLOOMFIELD

(45) a. f >> VP/NP + NP => VP
        f >> (VP\VP)/VP + VP => VP\VP
        ----------------------------------------
        Rff >> (VP\VP)/VP + VP/NP => (VP\VP)/NP

     b. He implements most changes [without consulting _ and without
        informing _] THE EXECUTIVE
```

In (46b) we Right Node Raise a noun phrase out of a complementized sentence.

```
(46) a. f >> S'/S + S => S'
        f >> S/NP + NP => S
        ----------------------------
        Rff >> S'/S + S/NP => S'/NP

     b. He said [that John met _ and that Mary saw _] SEVERAL
                                          FAMOUS LINGUISTS
```

Now one further **Rff** rule enables us to left extract out of embedded sentences:

```
(47) a. f >> VP/S' + S' => VP
        f >> S'/NP + NP => S'
        -------------------------------
        Rff >> VP/S' + S'/NP => VP/NP

     b. The modern dancing John   said     that    he   likes
        ------------------ ----  --------  ----   ----  ----------
               NP           NP  (S\NP)/S'  S'/S    NP   (S\NP)/NP
                                                   -------------Rfb
                                                          S/NP
                                           ----------------Rff
                                                  S'/NP
                                 ---------------------Rff
                                           (S\NP)/NP
                            ---------------------Rfb
                                     S/NP
                 -----------------------t
                             S
```

The three Right Node Raising rules we use in (47b) in fact enable extraction from any depth of embedding: the same subderivation can be repeated an arbitrary number of times thus:

```
1  Rfb >> NP + (S\NP)/NP => S/NP
2  Rff >> S'/S + S/NP => S'/NP
3  Rff >> (S\NP)/S' + S'/NP => (S\NP)/NP

NP (S\NP)/S' S'/S NP (S\NP)/S' S'/S NP (S\NP)/S' S'/S NP (S\NP)/NP
                                                         -----1------
                                                            S/NP
                                                   ------2------
                                                       S'/NP
                                        ---------3---------
                                             (S\NP)/NP
                                   ---------1---------
                                         S/NP
                            -------2-------
                                 S'/NP
                   ----------3----------
                         (S\NP)/NP
             --------1---------
                    S/NP
      --------2-------
            S'/NP
 ---------3-----------
      (S\NP)/NP
---------1---------
     S/NP
```

Following Dowty (1978) and Bresnan (1982) we assume that local phenomena such as dative shift, control and passivization are lexical. For example the lexical entry (lent, (VP/NP)/NP, lent$'_2$) for *lent* as in *I lent John a book* will be derived from that for *lent* as in *I lent a book to John*, (lent, (VP/(VP\VP))/NP, lent$'_1$), in such a way that lent$'_2$yx =

$lent'_1x(to'_{adv}y)$. Similarly the passive lexical entry (destroyed, VP, $destroyed'_{pas}$) will be related to the active lexical entry (destroyed, VP, $destroyed'_{act}$) by $were'(by'_{adv}x destroyed'_{pas})y = destroyed'_{act}yx$. It may be possible to characterise control phenomena such as that in (48b), where the matrix subject is also the subject of the verb phrase complement, by 4-leaf metarules such as ([*[[**]*]*], 2, 4, \).

(48) a. I want John to go
b. I want to go

However we believe that control phenomena are lexical so that *want* in (48a), (want, (VP/VP)/NP, $want'_1$), and *want* in (48b), (want, VP/VP, $want'_2$), will be related by $want'_2yx = want'_1xyx$.

The second metarule is called Left Node Raising. Left Node Raising, defined as ([[**]*], 2, 3, \), says that if there is a left-branching derivation path with mother V and leaves X, Y and Z in left-to-right order, then there is a rule with left-hand daughter Y, right-hand daughter Z, and mother V\X. Left Node Raising is the mirror-image of Right Node Raising. In Jowsey's notation the metarule is (49). Note that unlike Right Node Raising it is unconstrained.

```
(49)    φ >> X + Y => W
        ψ >> W + Z => V
        -------------------
        Lφψ >> Y + Z => V\X
        where Lφψ = λyλzλx[ψ(φxy)z]
```

The semantics of the combinator **L** arises as follows. The semantics of the input derivation path is $\lambda x\lambda y\lambda z[\psi(\phi xy)z]$. The semantics of the output rule is $\lambda a\lambda b\lambda c[\chi c^{j-1}ac^{k-j-1}bc^{n-k}]$ where χ is the semantics of the input. In this case j = 2, k = 3 and n = 3 therefore the semantics of the output rule is $\lambda a\lambda b\lambda c[\lambda x\lambda y\lambda z[\psi(\phi xy)z]cab] = \lambda a\lambda b\lambda c[\psi(\phi ca)b]$.

Again our first instances give rise to constructions which Schachter and Mordechay (1983) have already dubbed Left Node Raising:

```
(50)     f >> (VP/NP)/NP + NP => VP/NP
         f >> (S\NP)/NP + NP => S\NP
         -------------------------------------
         Lff >> NP + NP => (S\NP)\((VP/NP)/NP)

      b. He LENT [ _ John a book and _ Mary a paper]

      c. He    lent         John       a book
         --  -----------    ----       ------
         NP  (VP/NP)/NP      NP          NP
                         -------------------Lff
                         (S\NP)\((VP/NP)/NP)
             --------------------------------b
                           S\NP
         ------------------------b
                     S

(51) a.  f >> (VP/(VP\VP))/NP + NP => VP/(VP\VP)
         f >> VP/(VP\VP) + VP\VP => VP
         ----------------------------------------
         Lff >> NP + VP\VP => VP\(VP/(VP\VP))/NP)

      b. He GAVE [ _ a book to John and _ a paper to Mary]
```

Dowty (1985) also provides a combinatory account of Left Node Raising using backward composition and type-lifting operations which are parallel to Steedman's forward operations. The example (52b) is taken from Dowty's paper.

```
(52) a.  f >> (S/(VP\VP))/NP + NP => S/(VP\VP)
         f >> S/(VP\VP) + VP\VP => S
         ---------------------------------------
         Lff >> NP + VP\VP => S\((S/(VP\VP))/NP)

      b. [Bill gave and Max sold] [a book to Mary and a record
                                                     to Susan]

      c. Bill           gave                  a book    to Mary
         ----  -------------------            ------    -------
          NP   ((S\NP)/(VP\VP))/NP              NP       VP\VP
         ----------------------Rf(Rfb)    ------------------Lff
            (S/(VP\VP))/NP                S\((S/(VP\VP))/NP)
            --------------------------------------------------b
                                    S
```

Our next example is a hybrid construction to which Steedman's combinatory operations do not seem to extend. In the example there is simultaneous Left and Right Node Raising. Roughly speaking Steedman's operations encounter two problems. Firstly the second object seems to need to be type-lifted twice: once over the adnominal and once over the verb, but having type-lifted once, type-lifting again does not yield the appropriate category. Secondly, even if the appropriate category could be obtained, the category is not of the correct form to undergo backward composition, the operation which Dowty proposes for Left Node Raising. MCG captures the data through an instance of Right Node Raising one of whose input rules is itself produced by Left Node Raising:

```
(53) a. f >> NP/(N\N) + N\N => NP
        Lff >> NP + NP => (S\NP)\((VP/NP)/NP)
        ----------------------------------------------------------
        Rf(Lff) >> NP + NP/(N\N) => ((S\NP)\((VP/NP)/NP))/(N\N)

     b. He LENT [ _ John a book _ and _ Mary a paper _] ABOUT
                                                   SUBJACENCY

     c. He    lent        John             a book       about subjacency
        --  ----------    ----            --------     ----------------
        NP  (VP/NP)/NP    NP              NP/(N\N)          N\N
                          -------------------------Rf(Lff)
                          ((S\NP)\((VP/NP)/NP))/(N\N)
                          -------------------------------------f
                                      (S\NP)\((VP/NP)/NP)
            --------------------------------------------b
                                 S\NP
        ---------------------------------b
                          S
```

Left and Right Node Raising do not change word order. Our third metarule, Middle Node Raising, ([[**]*], 1, 3, /), does. The metarule takes the same left-branching inputs as Left Node Raising and produces rules whose daughters are the categories of the first and the third leaves and whose mother is C_0/C_r where C_0 was the category of the root and C_r was the category of the middle leaf.

```
(54)    φ >> X + Y => W
        ψ >> W + Z => V
        --------------------
        Mφψ >> X + Z => V/Y
        where Mφψ = λxλzλy[ψ(φxy)z] and Y ∈ MOVEABLE
```

The semantics of the input derivation path is $\lambda x\lambda y\lambda z[\psi(\phi xy)z]$ therefore the semantics of the output rule is $\lambda a\lambda b\lambda c[\lambda x\lambda y\lambda z[\psi(\phi xy)z]c^{1-1}ac^{3-1-1}bc^{3-3}] = \lambda a\lambda b\lambda c[\psi(\phi ac)b]$. The instance of Middle Node Raising in (55a) admits heavy noun phrase shift such as that exhibited in (55b)

```
(55) a. f >> (VP/(VP\VP))/NP + NP => VP/(VP\VP)
        f >> VP/(VP\VP) + VP\VP => VP
        ----------------------------------------
        Mff >> (VP/(VP\VP))/NP + VP\VP => VP/NP

     b. I put _ on the table A VALUABLE MING VASE WHICH I
                                 INHERITED FROM AUNT MAUD
```

We can already analyse a determiner-noun sequence as a noun phrase type-lifted over an adnominal. Middle Node Raising allows this adnominal to be right extraposed past a verb phrase:

```
(56) a. f >> NP/(N\N) + N\N => NP
        b >> NP + S\NP => S
        ---------------------------------
        Mfb >> NP/(N\N) + S\NP => S/(N\N)

     b. A man _ arrived WHO SPOKE RUSSIAN

     c. A man      arrived who spoke Russian
        --------   ------- ----------------
        NP/(N\N)   S\NP          N\N
        -------------Mfb
             S/(N\N)
             ---------------------f
                      S
```

Steedman's combinatory operations can also capture right extraposition via a slash-mixing backward composition operation. However they do not allow a verb phrase and a right extraposed adnominal phrase to form a constituent so that they fail to capture example (57b). MCG captures this through an instance of Left Node Raising which is recursive on Middle Node Raising:

```
(57) a. Mfb >> NP/(N\N) + S\NP => S/(N\N)
        f >> S/(N\N) + N\N => S
        ---------------------------------------
        L(Mfb)f >> S\NP + N\N => S\(NP/(N\N))

     b. (Around that time) NUMEROUS STATUES _ [ _ were des-
        troyed OF THE OLD PRESIDENT and _ were erected OF THE
        NEW ONE]

     c. Numerous statues were destroyed of the old president
        ---------------- -------------- ------------------
            NP/(N\N)          S\NP             N\N
                              -------------------L(Mfb)f
                                    S\(NP/(N\N))
           -----------------------------------b
                              S
```

We have shown how a determiner-noun sequence can analyse as a noun phrase doubly type-lifted over a subcategorized adnominal and a non-subcategorized one. In (57) we saw how a verb phrase and a right extraposed adnominal can form a constituent type-lifted over a noun phrase itself type-lifted over the adnominal. In (58a) the second (subcategorized) adnominal is Middle Node Raised so that two subject modifiers can be extracted in (58b).

```
(58) a. f >> (NP/(N\N))/(N\N) + N\N => NP/(N\N)
        b >> NP/(N\N) + S\(NP/(N\N)) => S
        ------------------------------------------------------
        Mfb >> (NP/(N\N))/(N\N) + S\(NP/(N\N)) => S/(N\N)

     b. (This is a woman)
         about     whom           an    argument    started  which ...
        --------- -------------   ----- ---------   -------- ----------
        (N\N)/NP  (N\N)/(S/NP)    NP/N  N/(N\N)      S\NP     N\N
        ----------------------p   ---------Rf(Rbf)  ---------L(Mfb)f
        (N\N)/(S/(N\N))           (NP/(N\N))/(N\N)  S\(NP/(N\N))
                                  ------------------------------Mfb
                                              S/(N\N)
        --------------------------------------------------f
                             N\N
```

Our final metarule ([[**][**]], 1, 3, /), the one underlying parasitic extractions, is called Parasitic Node Raising:

```
(59)    ϕ >> X + Y => W
        ψ >> Z + Y => U
        χ >> W + U => V
        --------------------
        Pϕψχ >> X + Z => V/Y
        where Pϕψχ = λxλzλy[χx(ϕxy)(ψzy)] and Y ∈ {NP}
```

The semantics of the input derivation is $\lambda x\lambda y\lambda z\lambda u[\chi(\phi xy)(\psi zu)]$. Therefore the semantics of the output rule is $\lambda a\lambda b\lambda c[\lambda x\lambda y\lambda z\lambda u[\chi(\phi xy)(\psi zu)]c^{1\text{-}1}ac^{3\text{-}1\text{-}1}bc^{4\text{-}3}] = \lambda a\lambda b\lambda c[\lambda x\lambda y\lambda z\lambda u[\chi(\phi xy)(\psi zu)]acbc] = \lambda a\lambda b\lambda c[\chi(\phi ac)(\psi bc)]$. The instance of Parasitic Node Raising in (60a) enables the parasitic extraction in (60b).

```
(60) a. f >> VP/NP + NP => VP
        f >> ((S\NP)\VP)/NP + NP => (S\NP)\VP
        b >> VP + (S\NP)\VP => S\NP
        ---------------------------------------------
        Pffb >> VP/NP + ((S\NP)\VP)/NP => (S\NP)/NP

     b. (This is a town) WHICH I bought a ticket to _ not wanting
                                                       to visit _

     c. bought a ticket to        not          wanting  to    visit
        --------------------  ---------------  -------  ----- -----
             VP/NP            ((S\NP)\VP)/VP   VP/VP    VP/VP VP/NP
                                                        -----------Rff
                                                           VP/NP
                                               ----------------Rff
                                                    VP/NP
                              ---------------------------Rff
                                   ((S\NP)\VP)/NP
            ---------------------------------------------Pffb
                          (S\NP)/NP
```

An instance of Left Node Raising which is recursive on both Right Node Raising and Parasitic Node Raising allows a verb complement and an adverbial phrase from which there is parasitic extraction to form a constituent:

```
(61) a. Rff >> VP/NP + NP/NP => VP/NP
        Pffb >> VP/NP + ((S\NP)\VP)/NP => (S\NP)/NP
        ---------------------------------------------------------------
        L(Rff)(Pffb) >> NP/NP + ((S\NP)\VP)/NP => ((S\NP)/NP)\(VP/NP)

     b. This is a town WHICH I BOUGHT [ _ a ticket to _
                                  not wanting to visit _
        and _ a ticket from _ not wanting to leave _]

     c. bought a ticket to not wanting to visit
        ------ ----------- -------------------
        VP/NP     NP/NP         ((S\NP)/VP)/NP
                  ---------------------------L(Rff)(Pffb)
                       ((S\NP)/NP)\(VP/NP)
        ----------------------------------b
                     (S\NP)/NP
```

The grammar we have used in this section is defined in its entirety in the appendix. We have outlined some of the phenomena for which this grammar provides an account, concentrating on constructions to which many other theories do not readily extend. The acceptability of such data is not always high, and it could of course be claimed that some of the data might be ungrammatical. Our philosophy, which we sketch in the next section, is one that favours explanations of quasi-acceptability in terms of the grammar itself.

3. Discussion

We begin this section with a discussion of competence and performance, and acceptability and grammaticality. We go on to compare MCG metarules with GPSG metarules and transformations.

Here, we view the problem of describing a grammar in a language system as analogous to the problem of describing a subroutine in a computer program. On the one hand, a subroutine has as a property that which it can in principle do, so for example a subroutine may compute the result of raising its first argument to the power of its second for arbitrary arguments. On the other hand there is that which the subroutine actually ever does, thus when a program runs, a subroutine will only ever be called on certain argument values. The first property is the subroutine's potential, the second is that subset of its potential which it ever actually exerises. The subset of the potential exercised is determined by the rest of the program, and also the subroutine itself in so far as it influences the rest of the program. Imported to linguistics, the dichotomy is that between competence and performance: there is that which a module (subroutine or component) is competent to do, and there is that which it ever actually performs. More precisely we can define the **competence** of a module as that set of arguments for which the module could compute a result, and the **performance** of a module as that set of arguments for which a module ever does compute a result.

It is only exceptionally that the competence and performance of a module coincide and for a module such as a grammar with an infinite domain they can never do so, hence the grammaticality-acceptability non-correspondence that we see in say, centre embedding.

It seems that there must be grammatical expressions which are not acceptable. Now consider whether there may be expressions which are acceptable but not grammatical. The problem is that if we entertain this possibility then the set of grammatical expressions will be neither a subset nor a superset of the acceptable expressions, so that the notion of grammaticality becomes non-predictive and empirically irrefutable: an expression may be acceptable or unacceptable irrespective of whether or not it is grammatical. To avoid this vacuity we advocate the **monotonicity hypothesis**: the hypothesis that the failure of any one component cannot be over-ridden; an immediate corollary of the hypothesis is that all acceptable expressions are grammatical. A further consequence is that if an expression is semi-acceptable, it must be grammatical.

If we are not going to attribute intermediate acceptability to ungrammaticality, then it falls at least in part to the grammar to account for quasi-acceptability. MCG goes some way towards doing this in virtue of the fact that expressions are generated at particular orders. Hence, other things being equal, a meta-categorial grammar predicts that expressions involving higher-order rules will not be more acceptable than ones involving lower order ones. The cases where we can be most sure that other factors are constant are those in which a string is ambiguous. Consider for example (62).

(62) A review of a book just came out which Chomsky wrote

The preferred reading is the one where the right extraposed relative clause modifies *review* rather than *book*. Under MCG this fact receives the explanation that the analysis of *a review of a book* as a noun phrase type-lifted over an adnominal modifying *review* is first-order, while the analysis of it as one type-lifted over an adnominal modifying *book* is second-order. Similarly the dominance of one reading of the topicalised

(63) On Monday I saw the girl who lifeguards!

is explained by the fact that the analysis of ***I saw the girl who lifeguards*** as a sentence type-lifted over an adverb modifying *saw* is first-order, while its analysis as a sentence type-lifted over an adverb modifying *lifeguards* is third-order.

While MCG will transgress many traditional constraints at higher orders, some will always be respected. For example the constrained meta-categorial grammar we have defined respects the Fixed Subject Constraint (Bresnan 1972) so that it never generates expressions such as

(64) *(This is the man) WHO we believe that _ spies

The reason for this is as follows. Since left-extraction is derivative on right-extraction and the extraction site is not clause-final, the only (non-parasitic) metarule that changes word order, Middle Node Raising, would have had to have applied for the extraction to take place. The input to Middle Node Raising has structure [[**]*]. Canonically S'/S-NP-VP has structure [*[**]] so Middle Node Raising cannot apply; this is the right form for Right Node Raising to apply but verb phrases are not members of MOVEABLE and so it cannot. Consequently the structure [*[**]] of complementized sentences is inviolate and the subject noun phrase cannot move.

Since MCG metarules resemble both transformations and GPSG metarules it is of interest to consider exactly where they differ from these antecedents. We will classify the three kinds of operation according to their applicability, the nature of their input, and the nature of their output. MCG metarules can apply to derivation paths projected by rules which are themselves the outputs of metarules, and transformations can apply to (terminated) derivations which are the outputs of transformations, so MCG metarules and transformations are both recursive in their applicability. GPSG metarules however are not. If this restriction were to be relaxed, it would also be necessary to allow additional categories to appear on the GPSG slash feature. Head-driven Phrase Structure Grammar (HPSG; Pollard 1985a,b) is an extension of GPSG allowing such stack-valued features. The kinds of metarules we are proposing are consistent with theories like HPSG because any theory allowing stack-valued features can model Categorial Grammar.

The inputs to MCG metarules are derivation paths. As we have seen we can equivalently view MCG metarules as applying directly to sets of rules since the set of derivation paths which a set of rules projects is implicit in the set of rules. GPSG metarules apply to a single rule though interestingly Gazdar et al. (1982) present what is in effect a metarule taking two input rules. Transformations apply to terminated derivations (ones whose leaves are labelled with terminal symbols). Note that we cannot view metarules as likewise applying to terminated derivations because there is nowhere in the output for material dominated by the node that is raised. Finally, the outputs of GPSG and MCG metarules are rules and the outputs of transformations are, of course, terminated derivations. Our classification is summarized in figure 3.

Despite these differences between transformations and MCG metarules, the analogy between a meta-categorial grammar's base grammar and metarules, and a transformational grammar's base component and transformations makes it natural to wonder whether there is any difference between between our meta-categorial grammar containing Right, Left, Middle and Parasitic Node Raising, and a transformational grammar whose base is the same as the base grammar of the meta-categorial grammar, and whose transformations are the analogues in (65).

(65) a. RNL: [A [B C]] --> [[A B] C]
b. LNL: [[A B] C] --> [A [B C]]
c. MNL: [[A B] C] --> [[A C] B]
d. PNL: [[A B] [C B]] --> [[A C] B]

In relation to this question consider again the example in (66).

(66) He LENT [_ John a book and _ Mary a paper _] ABOUT SUBJACENCY

Whatever coordination rule we might have, we assume that in order to capture (66) it must be possible to analyse *John a book* in (67) as a constituent.

(67) He lent John a book about subjacency

Consider how this might be done through cyclic application of the transformations in

Operation	Application	Input	Output
transformation	recursive	terminated derivation	terminated derivation
GPSG metarule	non-recursive	rule	rule
MCG metarule	recursive	derivation path (group of rules)	rule

Figure 3: Comparison of transformations, GPSG metarules, and MCG metarules

(65). The base or deep structure is [I [[gave John] [a [book [about subjacency]]]$_{NP}$]]. Applying RNL at the labelled node we obtain [I [[gave John] [[a book] [about subjacency]]]$_{VP}$]. Then we can apply at the labelled node either LNL to yield [I [gave [John [[a book] [about subjacency]]]]$_{VP}$] or RNL to yield [I [[[gave John] [a book]] [about subjacency]]$_{VP}$]. But now we cannot RNL *about subjacency* in the former without bringing *a book* too, and we cannot LNL *gave* in the latter without lifting *John*. Thus unlike the meta-categorial grammar, the corresponding transformational grammar does not analyse *John a book* in (67) as a constituent and so the grammars do not give rise to the same constituents. Note incidentally that we were applying transformations at a node, VP, at which transformations do not normally apply. We suspect that if the restriction to cyclic application above were lifted then the transformations would indeed generate the same constituents as the meta-categorial grammar; however one distinction would still remain. We said in the introduction that we assume that conjuncts such as *John a book* in *He lent John a book and Mary a paper about subjacency* form meaningful subexpressions. Applying transformations to the canonical derivation of *He lent John a book about subjacency* may indeed produce a tree in which *John a book* is dominated by a single node, but it would not assign any meaning to that subexpression.

We conclude with a couple of comments on MCG and universal syntax. As we have defined MCG there are 72 possible metarules, the majority of which are parasitic. To see this recall that a metarule is a quadruple $(T, j, k, |)$ where T is an $n(= 3$ or $4)$-leaf binary tree, $1 \leq j < k \leq n$, and $|$ is either / or \. There are two 3-leaf binary trees, [[**]*] and [*[**]], and five 4-leaf binary trees, [[**][**]], [*[[**]*]], [*[*[**]]], [[[**]*]*], and [[*[**]]*]. When $n = 3$, the number of ways of assigning j and k such that $1 \leq j < k \leq n$ is 3, and when $n = 4$ it is 6. | may take one of two values thus there are $2 \times 3 \times 2 = 12$ 3-leaf metarules and $5 \times 6 \times 2 = 60$ 4-leaf metarules. Our conjecture is that in non-configurational languages free word order is a product of order-changing metarules and perhaps order-changing lexical rules, canonical word orders rapidly becoming obscured by the interaction of unconstrained order-changing metarules.

Appendix: Constrained Meta-Categorial Grammar

In this appendix we define Constrained Meta-Categorial Grammar in which metarules are supplied with constraints on the node being raised. We also formally define the constrained meta-categorial grammar which was the sample grammar of section 2.

A **constrained metarule** is defined to be a quintuple $(T, j, k, |, \Gamma)$ where T is an $n(= 3$ or 4)-leaf binary tree, $1 \leq j < k \leq n$, | is / or \, and Γ is a set of categories, possibly the universal set of categories indicated by U. Then a **constrained meta-categorial grammar** is a quadruple $(\Delta, \Gamma, \Pi, \Sigma')$ where Δ is a set of basic types, Γ is a set of categories which is a subset of Cats(Δ), Π is a set of rules which is a subset of Rules(Δ, Γ), and Σ' is a set of constrained metarules. A constrained meta-categorial grammar $(\Delta, \Gamma, \Pi, \Sigma')$ defines a hierarchy of combinatory grammars $CMCG_0(\Delta, \Gamma, \Pi, \Sigma')$, $CMCG_1(\Delta, \Gamma, \Pi, \Sigma')$, $CMCG_2(\Delta, \Gamma, \Pi, \Sigma')$, ... as follows:

(A1) $CMCG_i(\Delta, \Gamma, \Pi, \Sigma') = (\Delta, \Gamma_i, \Pi_i \cup \Pi_i')$ where

$\Pi_0 = \{\}$,

$\Pi_{i+1} = \Pi \cup \{(\lambda a \lambda b \lambda c[sc^{j-1}ac^{k-j-1}bc^{n-k}], C_j, C_k, C_0|C_r)$:

$(T, j, k, |, \Gamma') \in \Sigma'$,

$(T, (C_r^{j-1}, C^j, C_r^{k-j-1}, C_k, C_r^{n-k}), C_0, \phi) \in$ Paths$(\Delta, \Gamma_i, \Pi_i \cup \Pi_i')$,

and $C_r \in \Gamma'\}$,

$\Gamma_0 = \Gamma$,

$\Gamma_{i+1} = \Gamma_i \cup \{C: (\phi, C_1, C_2, C) \in \Pi_{i+1}\}$, and

$CG(\Delta, \Gamma_i) = (\Delta, \Gamma_i, \Pi_i')$

The constrained meta-categorial grammar we have explored in this paper is specified in (A2).

(A2) ({S, S', N, NP},
{S, S\NP, (S\NP)/NP, ((S\NP)/NP)/NP, (S\NP)/S', (S\NP)/((S\NP)/(S\NP)), ((S\NP)/((S\NP)/(S\NP)))/NP, (S\NP)/(S\NP), (S\NP)\(S\NP), ((S\NP)\(S\NP))/NP, ((S\NP)\(S\NP))/(S\NP), S', S'/S, N, N/(N\N), N/N, N\N, (N\N)/NP, (N\N)/(S\NP), (N\N)/(S/NP), NP, NP/N},
{(λxλy[yx], NP, S/NP, S),
(λxλy[yx], S', S/S', S),
(λxλy[yx], N\N, S/(N\N), S),
(λxλy[yx], (S\NP)\(S\NP), S/((S\NP)\(S\NP)), S),
(λxλyλz[y(λw[z(xw)])], NP/NP, (N\N)/(S/NP), (N\N)/(S\NP))
(λxλyλz[y(λw[z(xw)])], NP/NP, (N\N)/(S/NP), (N\N)/(S/NP)),
(λxλyλz[y(λw[z(xw)])], (N\N)/NP, (N\N)/(S/NP), (N\N)/(S/(N\N))),
(λxλyλz[y(λw[z(xw)])], ((S\NP)\(S\NP))/NP, (N\N)/(S/NP), (N\N)/(S/((S\NP)\(S\NP))))},
{(([*[**]], 1, 2, /), {S', NP, N\N, (S\NP)\(S\NP)}),
(([[**]*], 2, 3, \), U),
(([[**]*], 1, 3, /), {S', NP, N\N, (S\NP)\(S\NP)}),
(([[**][**]], 1, 3, /), {NP})})