MARK STEEDMAN

COMBINATORY GRAMMARS AND PARASITIC GAPS*

The following sentence exemplifies what Taraldsen (1979) and Engdahl (1981, 1983) have christened 'parasitic gap' constructions, in which a single extracted item is (unboundedly) dependent upon more than one element of the matrix sentence.[1]

(1) Which articles did you file without reading?

Such multiple unbounded dependencies present interesting problems for any theory of grammar, and in particular for the one proposed by Ades and Steedman (1982, hereafter A&S) and Steedman (1985, hereafter D&C). These papers argued that a wide variety of extractions, discontinuous and coordinate constructions could be captured by adding operations of functional composition to the Categorial Grammar (CG) proposed by Ajdukiewicz (1935) Bar-Hillel (1953) and others. The

* This paper arises as part of a more ambitious program of work with Anna Szabolcsi. It began life as my draft of a paper to be published jointly with her, but has turned out to be rather more specialised in its concerns than we had anticipated, so that it now seems more appropriate to us that it should be published separately. Since our two quite distinct drafts of this earlier paper circulated under the title 'Combinators, Categorial Grammar, and Parasitic Gaps', I have slightly changed the title of the present paper in an attempt to avoid confusion with either draft. Nevertheless, the present paper remains directly influenced at a number of points by Anna's work and her comments on my unreasonably proliferating drafts, as I hope I have made clear in the text. It has also benefitted from comments over a long period by: Peter Buneman, Robin Cooper, Annabel Cormack, Kit Fine, Mark Gawron, Gerald Gazdar, Nick Haddock, Frank Heny, Jack Hoeksema, Polly Jacobson, Aravind Joshi, Einar Jowsey, Ewan Klein, David McCarty, Glyn Morrill, Dick Oehrle, Mats Rooth, Remo Pareschi, Ivan Sag, Mary Tait, Henry Thompson, Bonnie Lynn Webber, and the referees for *NLLT*. Some parts were presented at the Conference on Categorial Grammar at the University of Arizona, Tucson, June 1985 and the Conference on Modularity at the University of Massachusetts and Hampshire College, Amherst, June 1985, and I am grateful to the participants for their remarks and criticism. Parts of the research were supported by grants from ESPRIT (project 393) to CCS, University of Edinburgh, from the Cognitive Science Program at the University of Pennsylvania sponsored by the Alfred P. Sloan Foundation, and from NSF grant IRI-10413 A02, ARO grant DAA6-29-84K-0061 and DARPA grant N0014-85-K0018 to CIS, University of Pennsylvania.

[1] Terms like 'extraction', 'coordination reduction', 'gap' and other drawn from the technical vocabulary of transformational grammar are used purely descriptively in the present paper.

Natural Language and Linguistic Theory **5** (1987) 403–439.

present paper revises and extends the theory of unbounded dependencies to include those in parasitic gap constructions.

Functional composition is a very simple example of a class of operations on functions and arguments called "combinators", which were proposed by Curry and Feys (1958) in order to define the class of "applicative systems" that includes the lambda calculi. (An applicative system is simply a calculus which defines the notions of application of a function, and functional abstraction – that is, definition of a function in terms of some other(s).) The syntax, and particularly the semantics, of natural constructions like the relative clause in (2a) is strongly reminiscent of the lambda abstraction in (2b):

(2)a. (a man) whom Harry liked
b. λx ((LIKE x) HARRY)

Indeed, most current linguistic theories implicitly or explicitly assume that extraction is related to variable binding, and that the underlying form of (2a) actually is something like (2b). However, it is striking that there seem to be no explicit linguistic entities in (2a) corresponding to the variable-binding lambda operator and the bound variable itself. It is therefore interesting to ask whether other less familiar applicative systems might have a more transparent relation to such expressions in natural language. The interesting feature of the combinators in this connection is that they allow us to define the equivalent of abstraction using operations that are entirely local, and operate only on adjacent, linguistically realised, entities, without the use of bound variables.

The present proposal adds combinatory rules of a kind first proposed for natural language syntax by Szabolcsi (1983). A companion paper to the present one (Steedman, 1988, hereafter C&G) shows that such rules correspond semantically to another combinator which functions as an important primitive in Curry's account of theoretical foundations of the lambda calculus. The theory therefore holds out the promise of a very transparent relation between the syntax and the semantics.[2] C&G discusses in greater depth the nature of combinatory logic, and the mathematical and computational implications of the related syntactic combinators for natural grammars. The present paper investigates more purely linguistic implications.

[2] Curry himself followed Ajdukiewicz and others in suggesting the existence of a close link between applicative systems and natural language syntax (see Curry and Feys, 1958, pp. 274–75; Curry, 1961).

The paper begins by briefly reviewing the proposal to augment categorial grammars with functional composition, revising the earlier analysis of constituent order, coordinate structure, and extraction in almost every detail. The present incarnation of the theory is improved in several important respects by the inclusion of *directional* lexical categories, as previously proposed by Lambek (1958) Lyons (1968) and others. A number of claims are made concerning the possible combinatory rules that are available to Universal Grammar, and a number of constraints on single dependencies are shown to follow. The principal focus of the paper involves parasitic gap and it is shown that the same constraints on possible rules hold in the domain of multiple dependencies.

1. COMBINATORY GRAMMARS

Categorial grammars consist of two components. The first is a categorial lexicon, which associates each word of the language with at least one syntactic category, and distinguishes between *functions*, like verbs, and their *arguments*. The second component is a set of rules for *combining* functions and arguments, which are called *combinatory rules* because of their close relation to Curry's combinatory logic. In the original categorial grammar of Ajdukiewicz, this component was restricted to rules of functional application, and made the grammar context-free, equivalent to the more familiar phrase-structure grammars. Later versions (Lambek, 1958, 1961; Geach, 1972; Bach, 1979, 1980; A&S; Huck, 1985; Oehrle, 1988) have included more complex combinatory operations. However, all of these extensions conform to the following limiting principle, as does the original operation of functional application of Ajdukiewicz:

(3) *The Principle of Adjacency*
Combinatory rules may only apply to entities which are linguistically realised and adjacent.

1.1. *The Categorial Lexicon*

Some syntactic categories, such as N, the category of nouns, are atomic symbols. In the present theory, functions which combine with arguments to their right bear a category of the form X/Y, denoting a rightward-combining function from category Y into category X. For example, determiners are NP/N transitive verbs are VP/NP. Other functions which combine with their arguments to the left are distinguished by the use of a backward slash, and a category of the form $X\backslash Y$, denoting a leftward-combining function from Y into X.[3] For example, VP-adverbial phrases like *quickly* bear the category VP\VP, and predicate phrases like *arrived* bear the category S\NP. (The reader is warned that some authors, including Lambek and Bach use an opposite convention, under which this function would be written "NP\S", with the argument on the left. This alternative is less readable with the multiple argument function categories used in the present theory.)

Both types of function may of course have more than one argument, and may mix the two types of slashes, combining with different arguments in different directions. However, all function categories are unary or 'curried'. For example, the ditransitive verb *give* will bear the category (VP/NP)/NP – a (rightward-combining) function from (indirect) objects into (rightward combining) functions from (direct) objects into VPs.[4]

The combinatory rules govern the combination of such functions with adjacent arguments and with other functions. The first and simplest of these is a rule which applies a function to an argument.

1.2. *Functional Application*

On the assumption that syntactic categories directly reflects the semantics of the entity in question in a "type driven" fashion (cf. Klein and Sag, 1984), we can write the syntactic and semantic combinatory rules in one, associating each syntactic category in the rule with a semantic interpretation. (In this case, the semantics is trivial, but it serves to introduce some notation.)

(4) *Functional Application*
A function of category X/Y or $X\backslash Y$ and interpretation F can combine with an adjacent argument Y with interpretation y to yield a result of category X and interpretation Fy, the result of applying F to y.

In this and the other combinatory rules that follow, X and Y are variables which range over any category, including functions, so X/Y is any rightward-combining function and $X\backslash Y$ is any leftward-combining function. Upper case F, G, etc. are used for the interpretations of

[3] The present theory differs in this respect from its predecessors in A&S and D&C, which used *non*-directional slashes, constraining order in the combinatory rules.

[4] This restriction has no great significance. Unary nth order curried functions are equivalent to n-ary first order functions, as first noted by Schönfinkel (1924; cf. Dowty, 1982 for a brief discussion).

functions, while lower case x, y, etc. are used for the interpretations of arguments. The application of a function F to an argument x is represented by left to right order, as Fx.

The application of this rule to a function and an argument of the appropriate types is by definition subject to their left-to-right order being consistent with the directionality of the function, because that is what the slashes *means*. Obvious though this restriction is, it will be useful to state it explicitly under the title of the principle of DIRECTIONAL CONSISTENCY, as follows:

(5) *The Principle of Directional Consistency*
All syntactic combinatory rules must be consistent with the directionality of the principal function.

We will defer definition of the adjective "principal". (See (14) below.)

The functional application rule, constrained as it is by (5), gives rise to two specific instances. These are shown in (6), where in (6a) a right word combining function occurs to the left of a potential argument, to which it is applied while in (6b) a left word combining function occurs to the right of a potential argument, to which it is applied.

(6)a. $X/Y{:}F \quad Y{:}y \Rightarrow X{:}Fy$ (>*apply*)
b. $Y{:}y \quad X\backslash Y{:}F \Rightarrow X{:}Fy$ (<*apply*)

(Semantic interpretations appear to the right of syntactic categories, separated by a colon.) The first case, called FORWARD APPLICATION, allows rightward-combining functions like transitive verbs to combine with arguments to their right, as in the following derivations, in which the operation of combinatory rules is indicated by underlining the operands, indexing the underline with a mnemonic symbol (in this case >*apply*), and writing the result beneath.

```
(7)a.  Eat    the   cake      b.    Give       me  that
       -----  ----  ----         -----------   --  ----
       VP/NP  NP/N  N            (VP/NP)/NP    NP  NP
              ---------->apply   --------------->apply
                  NP                 VP/NP
       ---------------->apply        ------------------>apply
           VP                            VP
```

(Such diagrams are equivalent to the trees associated with phrase structure grammars.)

The second instance of the rule of functional application, (6b), allows a leftward-combining function $X\backslash Y$ to combine with an argument Y to its left. This instance of the functional application rule is indicated in derivations by an underline indexed by *<apply*. Not many function categories of English are backward-combining, but certain non-subcategorised-for adverbials are, as in:

```
(8)     Come   quickly
        ----   -------
         VP    VP\VP
        -------------- <apply
             VP
```

Such adverbials are discussed further in section 1.6 below.

1.3. *The Category of Subject and Verb*

Another apparent example of an argument occurring to the left of a function, and hence seeming to require backward application, is the subject of a sentence. It seems natural to assume that tensed verb phrases bear the category S\NP, so that tensed transitive verbs like *eat* are (S\NP)/NP, while ditransitives are ((S\NP)/NP)/NP and so on, giving rise to derivations like the following:

```
(9)     Harry     eats      apples
        -----  ----------   ------
         NP    (S\NP)/NP     NP
               ------------------- >apply
                      S\NP
        -------------------------- <apply
                  S
```

This derivation assigns an interpretation which we might write EAT′ APPLES′ HARRY′, where functional application associates to the left, so that the result is equivalent to ((EAT′ APPLES′) HARRY′). It is the interpretation of the verb, EAT′, which determines the grammacal relations of the first argument HARRY′ and the second, APPLES′, as subject and object, respectively.[5]

[5] We may note in passing that the last, subject, argument of the verb must be defined or plural or singular number by the inflection of the verb, and that the subject argument must be compatible with this specification, to capture basic subject verb agreement using a unification procedure of the kind proposed for this problem by Shieber (1986). For example, we might distinguish NPs by a feature, plural NPs being NPplr. Then $leave_{S\backslash NPplr}$ can combine with an NPplr, but not with NPsng. Other arguments of the verb will in English be unconstrained as to the value of this feature. An advantage of this approach is that, since the verb takes the whole clause as its "domain of locality" (Joshi, 1987), explicit feature passing through the VP node is unnecessary. In the remainder of the paper, we will usually not distinguish number in the categories and examples.

2. Coordination

Two central problems for any theory of natural language grammar are posed by coordination-reduction constructions, typified in (10a) below, and extraction, (typified in (10b):

(10)a. [I know Harry will cook] and [I think Betty might eat] the mushrooms we picked in the dank meadows behind the Grange.

b. These mushrooms, I think Betty might eat.

Both constructions appear to separate elements like objects and verbs which belong together semantically. Both may separate them by unbounded strings, including clause boundaries. They therefore appear to force us to abandon simple assumptions like the Principle of Adjacency (3), or the assumption that rules of grammar should apply to constituents. However, both of these phenomena can be analysed without abandoning either assumption, under some simple extensions to the combinatory rules and a consequent extension of the concept of a constituent to include entities corresponding to strings like *might eat* and *I think Betty might eat*.

2.1. *Functional Composition*

(11)

I	will	cook	and	might	eat
NP	(S\NP)/VP	VP/NP	conj	(S\NP)/VP	VP/NP

the mushrooms we picked
NP

Functional application will not help us here. But there is an almost equally simple operation which obeys the Principle of Adjacency, and which *will* help, namely functional composition.[6]

The combinator which composes two functions F and G is called **B** by Curry, and can be defined by the following equivalence:

(12) $\mathbf{B}FGx = F(Gx)$.

A convention that application associates to the left is again followed, so that the left hand side is equivalent to $((\mathbf{B}F)G)x$. It follows that we can consider the application of **B** to F and G as producing a new function

[6] The reader is directed to C&G for further discussion of the sense in which functional composition is simple. There is a precedent for the inclusion of rules of composition in CG in the work of Lambek (1958, 1961) and Geach (1972). See Wall (1972) for a brief introduction to the concept of functional composition.

equivalent to abstracting on x in the above expression, thus:

(13) $\mathbf{B}FG = [x]F(Gx)$.

Curry's bracket abstraction notation "$[x]$ ⟨expression⟩" means much the same as the lambda notation "λx ⟨expression⟩". We use it here to remind the reader that the combinators are the primitives not the abstraction operator.

It will be convenient to distinguish the two functions F and G in the above example as the PRINCIPAL and the SUBSIDIARY FUNCTION, respectively. Using this definition of functional composition we can state the following general combinatory rule:

(14) *Functional Composition*
A principal function over Y, of category X/Y or $X\backslash Y$ and interpretation F, may combine with an adjacent subsidiary function into Y of category Y/Z or $Y\backslash Z$ and interpretation G. The result is their syntactic and semantic composition, a function from Z into X of category X/Z or $X\backslash Z$ which bears the interpretation $\mathbf{B}FG$.

Like the rule of Functional Application (4), this rule is subject to the Principle of Directional Consistency (5): the subsidiary function must occur to whichever side is consistent with the slash on the principal function. The rule is also subject to a less obvious principle, which follows, I claim, from the semantics of the metalanguage, and limits all combinatory rules which produce a function as their output, as follows:[7]

(15) *The Principle of Directional Inheritance*
If the category that results from the application of a combinatory rule is a function category, then the slash defining directionality for a given argument in that category will be the same as the one defining directionality for the corresponding argument(s) in the input function(s).

The functional composition rule potentially gives rise to four instances, distinguished by the left to right order and directionality of the principal and subsidiary functions, as follows:

[7] The claim follows from the unification-based formalism which Pareschi (1986) and Pareschi and Steedman (1987) propose for combinatory grammar in an implementation of a parser.

(16)a. $X/Y{:}F \;\; Y/Z{:}G \Rightarrow X/Z{:}\mathbf{B}FG$ (>*compose*)
b. $X/Y{:}F \;\; Y\backslash Z{:}G \Rightarrow X\backslash Z{:}\mathbf{B}FG$ (>*xcompose*)
c. $Y\backslash Z{:}G \;\; X\backslash Y{:}F \Rightarrow X\backslash Z{:}\mathbf{B}FG$ (<*compose*)
d. $Y/Z{:}G \;\; X\backslash Y{:}F \Rightarrow X/Z{:}\mathbf{B}FG$ (<*xcompose*)

Natural languages are free to include rules on any of the four patterns, to restrict their application to certain categories, or to entirely exclude some of them. All four have been used to account for various phenomena in English.[8] The first that we shall consider here is (16a), in which the principal functor is on the left, and the slashes are all rightward, which will be referred to below simply as the Forward Composition Rule. The corresponding 'crossing' rule (16b) is excluded from the present grammar.

Consider the effect of this rule in the coordinate sentence (11) which this section set out to analyse. The categories of the adjacent functors $\textit{will}_{(S\backslash NP)/VP}$ and $\textit{cook}_{VP/NP}$ match the rule, as do the parallel categories *might* and *eat*. Given the simple coordination scheme (17), the sentence can be derived as in (18):

(17) Coordination
$X \;\; conj \;\; X \Rightarrow X$ (*coord*)

(18)

I will cook and might eat the mushrooms...
NP (S\NP)/VP VP/NP conj (S\NP)/VP VP/NP NP
——— >*compose* ——— >*compose*
(S\NP)/NP (S\NP)/NP
——— *coord*
(S\NP)/NP
——— >*apply*
S\NP
——— <*apply*
S

Thus, the combinatory rule given in (14) will, as promised, provide an analysis of (11).

2.2. *Type-raising*

Repeated application of Forward Composition to the verb sequences in examples like the following will allow coordination of indefinitely long

[8] Dowty (1988) has used the backward rule (16c) in his account of English non-constituent coordination. The slash-crossing backward rule (16d) is introduced below, and has also been used by Moortgat (1988) and Morrill (1987), while a very restricted version of the forward crossing rule (16b) is used in Steedman (1987) to account for gapping in English.

strings of verbs, on the assumption that each is a function over the result of the one to its right:

(19) She [may have seemed to have wanted to meet,]$_{(S\backslash NP)/NP}$
but [actually turned out to dislike,]$_{(S\backslash NP)/NP}$
the man you brought to the party.

However, one more combinatory rule must be included to accept related examples like the following:

(20)a. [I will cook] and [Betty may eat]
the mushrooms we picked in the dismal glens above the Grange.

b. [I think I will cook] and [you think that Betty may eat]
the mushrooms we picked . . . .

The problem with examples like (20a) is that the subject cannot combine with the tensed verb or the composed verb group, whose categories dictate that they have to combine with something else first:

(21)
$$\begin{array}{llll} \text{I} & \text{will} & \text{cook} & \ldots \\ \hline \text{NP} & \text{(S\textbackslash NP)/VP} & \text{VP/NP} & \\ & \multicolumn{2}{c}{\text{(S\textbackslash NP)/NP}} & >compose \\ \hline & & & * \end{array}$$

However, there is an operation which is widely used in the Montague Grammar literature to map arguments (such as subjects) into functions over functions-which-take-such-arguments (such as predicates). This operation is called "type-raising". Like composition, it has a simple and invariant semantics, and the general rule can be written as follows:

(22) *Type-raising*
A category Y with interpretation y can be replaced with a (higher-order) function category over functions X/Y or $X\backslash Y$ having interpretations F into a result of type X and an interpretation Fy, the result of applying F to y.

This rule is also assumed to be subject to the Principle of Consistency, which ensures a "direction preserving" property proposed by Dowty, such that arguments may only raise into *rightward*-looking functions over leftward looking ones, or into *leftward*-looking functions over rightward ones. The semantics corresponds to another of Curry's basic combinators, called $\mathbf{C}_*$, defined by the following equivalence:

(23) $\mathbf{C}_* xF = Fx.$

It follows that $\mathbf{C}_*$ applied to an argument creates the following abstraction over the function:

(24) $\mathbf{C}_* x = [F]Fx.$

There are two possible direction-preserving type-raising rules which we can write as follows:

(25) *Direction-preserving Type-raising* (*Unrestricted*)
a. $X{:}x \Rightarrow W/(W\backslash X){:}\mathbf{C}_* x$
b. $X{:}x \Rightarrow W\backslash(W/X){:}\mathbf{C}_* x.$

We restrict these rules to raise only when there is an adjacent function of an appropriate type, thus:[9]

(26) *Direction-preserving Type-raising* (*Restricted*)
a. $X{:}x \quad W\backslash X\${:}F \Rightarrow W/(W\backslash X){:}\mathbf{C}_* x \quad W\backslash X\${:}F \qquad (>\uparrow)$
b. $W/X\${:}F \quad X{:}x \Rightarrow W/X\${:}F \quad W\backslash(W/X){:}\mathbf{C}_* x \qquad (<\uparrow)$

The $ symbol on a category like $W/X\$$ means it denotes any member of the set Σ recursively defined as including W/X and all functions into members of the set.

Type raising is somewhat daunting notationally, but simply to apply in practice. The subject NP in example (21) is to the left of a function of the form $S\backslash NP\$$. It can therefore raise into the category $S/(S\backslash NP)$. This category can in turn compose with the verb, permitting the following derivation:

(27)

```
  I        will      cook   and  Betty    might      eat    the mushrooms ...
 ---  ----------- ------- ----  -----  ----------- ------- ---------------
 NP   (S\NP)/VP   VP/NP   conj   NP    (S\NP)/VP   VP/NP         NP
 ----- >↑                        ----- >↑
S/(S\NP)                        S/(S\NP)
------------------ >compose     ------------------ >compose
      S/VP                             S/VP
      ------------------- >compose     ------------------- >compose
              S/NP                             S/NP
              ---------------------------------------- coord
                              S/NP
                              ----------------------------------- >apply
                                               S
```

[9] Some such restrictions are tacitly assumed in earlier papers. We leave open the question of whether they should apply obligatorily or optionally to certain argument categories like NP. These rules alone will not cover all cases of type raising considered in the earlier papers and Dowty (1988).

The more complex example (20b) is accepted in a parallel manner, since the embedded subjects can also raise under the rule.

Dowty (1988), D&C, and Steedman (1987) extend the analysis to a wide range of 'non-constituent' coordination phenomena in Dutch and English.

3. Unbounded Dependencies

The two combinatory syntactic rules of functional composition and type raising provide all that we need in order to solve the second problem introduced in (10), that of leftward extraction in Wh-movement constructions. Thus, in the following example the raised subject category can again raise over the predicate, and iterated composition can again assemble the subject and all the verbs in the entire sequence *Harry must have been eating* to compose into a single function, thus:

(28)

```
These apples  Harry       must        have       been          eating
     NP       S/(S\NP)  (S\NP)/VP  VP/VPen   VPen/VPing   VPing/NP
              ---------------------->compose
                     S/VP
                     ------------------------>compose
                              S/VPen
                              --------------------------->compose
                                         S/VPing
                                         ----------------------------->compose
                                                      S/NP
```

The important result is that the entire clause has been assembled into a single function adjacent to the extracted argument. Technically, this function still cannot combine, because the directionality of the slash forbids it. There are a number of ways of handling this detail consistent with the Principle of Adjacency. I will use the following rule, which is related to type raising, but which is (obviously) *not* direction preserving, and which marks its result as "S_t" – an S marked with a feature, say [+TOPIC], preventing multiple topicalisation: Since the rule is not pure type raising, and its semantics is obscure, we omit the latter entirely.

(29) *Topicalisation*

$\# \; X \; \Rightarrow \; \# \; S_t/(S/X)$

where $X \in \{NP, PP, VP, AP, S'\}$.

(The boundary symbol # restricts the rule to the leftmost position in the sequence. The restriction on X is needed because not everything that we can right node raise out of can we leftward extract over. For example, compare *Harry and Barry went home* and **went home Harry and Barry*.)

Topics resulting from (29) can combine with the remainder of the sentence by the forward application rule, thus:

(30)

```
 Apples,     Harry      must      have       been       eating
St/(S/NP) S/(S\NP) (S\NP)/VP VP/VPen VPen/VPing VPing/NP
          ------------------- >compose
                 S/VP
                 ------------------- >compose
                        S/VPen
                        -------------------- >compose
                               S/VPing
                               -------------------- >compose
                                        S/NP
--------------------------------------------------- >apply
                        St
```

A similar analysis holds for relativisation as for topicalisation. I assume that restrictive relative clauses are noun adjuncts of type N\N. Object relative pronouns are then functions parallel to the topic category from S/NP into relative clauses N\N, written as follows:

(31) *who(m)* := (N\N)/(S/NP).

This category is again related to, but not the same as, a type raised category. I assume that it is assigned in the lexicon, not in free syntax. A simple relative clause is analysed as follows:

(32)

```
(apples)     which        Harry       eats
         (N\N)/(S/NP) S/(S\NP) (S\NP)/NP
                      ------------------ >compose
                             S/NP
         ---------------------------- >apply
                   (N\N)
```

The subject relative pronoun will bear the category (N\N)/(S\NP), allowing derivations like the following:

(33)

```
(a man)      who        left
        (N\N)/(S\NP) S\NP
        ------------------ >apply
              (N\N)
```

The rules of functional composition and type raising provide a general mechanism for unbounded extraction. On the assumption that one category for the tensed verb *believe* is (S\NP)/S′, and that the complementiser *that* is S′/S, repeated application of the forward composition rule allows extractions across clause boundaries:

(34)

```
  Apples,       I        believe   that   Harry      eats
 St/(S/NP)  S/(S\NP)  (S\NP)/S'   S'/S  S/(S\NP)  (S\NP)/NP
            ------------------- >compose
                   S/S'
                   ------------------- >compose
                          S/S
                          --------------------- >compose
                                S/(S\NP)
                                ------------------------- >compose
                                           S/NP
 ------------------------------------------------------ >apply
                     St
```

The theory predicts a conspiracy between the domains of left extraction and right node-raising. Both arise from the composition of the residue into a single function, so if something can left extract than it should be able to right node raise as well. In particular, there should be no independent "right roof constraint" (Ross, 1967), a condition which Gazdar (1981) has convincingly argued to be artifactual. The converse does not apply: the limitations on the topicalisable and relativised categories, and limitation of their domain to S mean that more categories can right node-raise than can undergo extraction, both within S and within NP (see D&C and Steedman 1987 for further discussion).

It should be clear by now that the theory implies a very unusual view of surface structure. If strings like *I believe that Harry eats* are constituents for the purposes of conjunction and extraction, then they must also be possible constituents of canonical sentences like *I believe that Harry eats these apples* as well. Moreover, for each reading of a sentence, there will typically be many different surface analyses, corresponding to different orders of applying composition and application. The consequences for processing seem potentially grave.

However, if we assume a level of interpretation which is neutral with respect to non-structure-dependent aspects of meaning, such as quantifier scope, then the associativity of functional composition ensures that all the derivations that arise from composing functions in different orders for a given set of function-argument relations will produce the *same* interpretation.[10] This fact both sanctions the coherence of the grammar itself, and points to a solution to the parsing problem: if these analyses are equivalent, it clearly doesn't matter *which* of them we find,

[10] This is not to say that composition will induce no new semantically distinct readings from a pure categorial grammar. The inclusion of the non-associative operation of application, together with the presence of higher-order functions, may actually induce new function argument relations.

so long as we find *one*. Pareschi (1986) and Pareschi and Steedman (1987) discuss an efficient and grammatically transparent solution to the processing problem based on this observation.

3.1. *Dependencies within adverbials*

Extraction is not possible from adjuncts. This follows automatically from the present theory, as does the fact that sub-categorised adverbials generally permit extraction. These cases are dealt with in turn.

3.1.1. *Non-subcategorised adverbials.* I continue to assume for ease of presentation that non-subcategorised-for adverbials like *quickly* bear the leftward-combining category VP\VP, ignoring the fact that they can often combine with other related verbal categories like S\NP, VPing, VPen, VPto-inf and the like. The categories and combinations that go to make up the verb phrase *file articles without reading them* are as follows:[11]

(35)
```
 file   articles     without        reading   them
------  --------  --------------  ----------  ----
VP/NP     NP      (VP\VP)/VPing   VPing/NP     NP
-------------->apply
      VP
                  ---------------------------->compose
                          (VP\VP)/NP
                          --------------------------->apply
                                     VP\VP
          ------------------------------------<apply
                          VP
```

The island status of the adverbial *without reading them/the articles* follows automatically from the fact that it is not an argument of the verb *file*:

(36)
```
*(articles)     which      I will  file letters  without reading
            ------------  ------  ------------  ---------------
            (N\N)/(S/NP)   S/VP        VP          (VP\VP)/NP
                                    ------------------------------ *
```

However, the categories and rules used so far in dealing with dependencies wrongly prevent the extraction of the object alone, as in:

[11] The nonce category VPing is semantically ⟨*e*, *t*⟩, like all VPs. I ignore the question of how the semantics can ensure that the semantic subject of the *-ing* complement is bound to the subject of the VP.

(37)

```
(letters)     which      I will   file    without telling  anyone
          (N\N)/(S/NP)  S/VP   VP/NP    (VP\VP)/NP        NP
                                        ----------------------- >apply
                                               VP\VP
                                ------------------------ ?
```

What we need is the fourth instance of the composition rule, (16d), repeated here (see Moortgat (1988) and Morrill (1987) for accounts of right extraposition using this rule):

(38) *Backward composition (crossing)*
$Y/Z{:}G \quad X\backslash Y{:}F \;\Rightarrow\; X/Z{:}\mathbf{B}FG$ (<*xcompose*).

(The rule must be further restricted in order to exclude sequences like *$the_{NP/N}$ $walks_{S\backslash NP}$ dog_{N}. The appropriate restriction seems to be to insist that Z be a major category. The derivation of example (37) then goes as follows:

(39)

```
(letters)     which      I will   file    without telling  anyone
          (N\N)/(S/NP)  S/VP   VP/NP    (VP\VP)/NP        NP
                                        ----------------------- >apply
                                               VP\VP
                                ------------------------ <xcompose
                                        VP/NP
                         ------------------------ >compose
                                 S/NP
          ------------------------------ >apply
                  (N\N)
```

This form of composition will of course allow considerable freedom of constituent order for adverbials. For example, it will, correctly, allow them to occur in Heavy NP Shifted constructions:

(40)

```
I will  destroy  without telling anyone  all articles more than five pages long
S/VP    VP/NP          VP\VP                          NP
        -------------------------- <xcompose
              VP/NP
------------------------ >compose
       S/NP
       ---------------------------------------------------------------- >apply
                                   S
```

(We ignore the question of why the object NP has to be 'heavy' here.) It will also allow related coordinations, such as:

(41)a. I will [destroy without telling anyone and forget utterly] all articles which are more than five pages long.

b. articles which I will [destroy without telling anyone and forget utterly].

The rule will not permit extraction from heavy shifted adverbials, so the grammar conforms to Kuno's (1973) "clause non-final incomplete constituent constraint" without further stipulation:

(42) *(articles) which I will eat without reading the apple which . . .

(N\N)/(S/NP) S/VP VP/NP (VP\VP)/NP NP

———————— *

Finally, the continued exclusion of forward slash-crossing composition (see the discussion of (16)) and the constraints imposed by the Principle of Inheritance upon Backward slash-crossing prevents overgeneralisations like the following (it is assumed here that the adverbial can be either a VP adverbial or a predicate adverbial):

(43) *He eat curry will without cooking rice

S/(S\NP) VP (S\NP)/VP VP\VP

———————— >*

(S\NP)\(S\NP)

———————————— <*

3.1.2. *Subcategorised Adverbials*. Since the composition mechanism disallows extraction from within backward-looking functions such as modifiers, it seems to follow that all prepositional phrases that permit extraction – that is, permit the preposition to 'strand' – must be arguments of the verb. Thus the following example seems to imply that the verb *put* has the category (VP/PP)/NP:

(44)

(a table) which I will put the book on

(N\N)/(S/NP) S/VP (VP/PP)/NP NP PP/NP

———————— >*apply*

VP/PP

———————— >*compose*

S/PP

———————————— >*compose*

S/NP

———————————————— >*apply*

(N\N)

However, the grammar as it stands will not permit the following unremarkable example:

(45) (books) which I will put on the table

(N\N)/(S/NP) S/VP (VP/PP)/NP PP

Such extractions of 'inner' arguments could be accommodated by generalising the composition rule and the raised category of preposed items (see e.g., Steedman, 1988). However, this device leaves unexplained the fact that substrings like *put on the table* follow exactly the same pattern under coordination and Heavy Shift as the adjuncts of the previous section. Thus:

(46)a. I will [put on my coffee table] this very rare and valuable book.

b. I will [purchase and put on my coffee table] this very rare and valuable book.

c. a book which I will [purchase and put on my coffee table].

The fact that fragments like *put on the table* can coordinate with transitive verbs like *purchase* suggests that they too are constituents of type VP/NP. In fact, the subcategorised adverbials behave in every respect like the non-subcategorised ones, as if they were backward-combining modifiers.

If PPs were type-raised under the direction-preserving rule (26b), to be VP\(VP/PP), then they could combine via the backward slash-crossing instance (38) of the composition rule to allow the problematic extraction (45), thus:

```
(47) (books)     which      I will     put       on the table
             ------------ ------ ----------- ------------
             (N\N)/(S/NP)  S/VP  (VP/PP)/NP  VP\(VP/PP)
                                 ------------------------<xcompose
                                           VP/NP
                          -------------------------->compose
                                      S/NP
             ------------------------------->apply
                          (N\N)
```

The same tactic will also allow the constructions in (46).[12]

3.2. *Constraints on Unbounded Dependencies*

In general, constraints on 'extraction' (or 'unbounded' dependencies) follow directly from the principles of the theory. This is shown to be the case for several constraints in this section.

---

[12] The Heavy shift and Rightward extraction derivations will not generalise to the subject, of course, because of the directionality of the predicate category.

3.2.1. *Island Constraints.* As already noted, the island status of adjuncts arises from their backward modifier category VP\VP. The Wh-island status of relative clauses, as enshrined in the Complex NP-constraint of Ross (1967), follows for exactly the same reason, because of the N-modifier category N\N of the relative clause.

However, the possibility of escaping from island constraints also exists in the theory, because type raising is allowed. For example, the following example, from Belletti via Chomsky (1982), is allowed if the VP can type raise under the rule (26b):

(48)

```
(books)     which     I will  go to London  without reading
        ------------- ------  ------------  ---------------
        (N\N)/(S/NP)   S/VP        VP          (VP\VP)/NP
                              ------------>↑
                               VP/(VP\VP)
                              ------------------------------ >compose
                                           VP/NP
                       ------------------------- >compose
                                  S/NP
        ----------------------------------- >apply
                     N/N
```

In a similar fashion, nouns, may potentially type-raise over noun modifiers, thus:

```
(49)   pictures     of
       --------  ---------
          N      (N\N)/NP
       ------->↑
       N/(N\N)
       -------------------- >compose
              N/NP
```

(Such raising might of course be done in the lexicon.) Within the present theory, therefore, these constraints merely express the fact that there is a cost associated with such type-raising. The fact that the constraints seem sensitive to semantic factors such as the identify of the verb, and the subjecthood and quantificational status of the NP, suggests that this cost is semantic in origin rather than syntactic, and reflects the relative 'reasonableness' of concepts like *seeing a picture of* and *a picture of being burned*.

3.2.2. *Subject Extraction.* Functional composition and type raising provide a general mechanism for unbounded extractions like the following (see the derivation in example (34)):

(50) Apples, I believe that Harry eats.

However, the corresponding *subject* extractions are excluded as a consequence of the English specific restriction against slash-crossing forward composition, so that the Fixed Subject Constraint or *that*-trace filter of Bresnan (1972) and Chomsky and Lasnik (1977) follows automatically:[13]

(51) *Harry | I believe | that | eats | apples

St/(S\NP) | S/S′ | S′/S | (S\NP)/NP | NP

———— >*compose*

S/S

———— >*apply*

S\NP

———— >**compose*

S\NP

———— >*apply*

St

The problem than arises of how to *allow* subject extraction examples like

(52) Harry, I believe eats apples.

It might seem that a simple way to achieve subject extraction would be to assume a further category VP/S for verbs like *believe*, and to allow the otherwise forbidden 'slash-crossing' forward composition into the predicate category, excluding (51) by a restriction on such composition into *S′*. This expedient would permit the following sort of derivation, closely related to the earlier examples of object extraction, except that the topic receives a category resembling that of a type-raised subject.

[13] The first proposal to derive the Fixed Subject Constraint from a restriction agair slash-crossing function composition was by Anna Szabolcsi, who like Dowty differed fro.. the present proposal in supposing that the prohibition was universal. I hope to discuss languages which escape the Fixed Subject Constraint in a paper which is in pre- -ion with her (Steedman and Szabolcsi, 1987).

(53)

```
 Harry      I believe   eats        apples
St/(S\NP)     S/S     (S\NP)/NP     NP
                      ----------------- >apply
                            S\NP
              ------------------- >*compose
                        S\NP
-------------------------- >apply
           St
```

However, the inclusion of composition on this pattern fails to acknowledge the exceptional character of subject extraction. It also threatens to allow overgeneralisations like the following:

(54)

```
*I  he   think    left
NP  NP  (S\NP)/S  S\NP
        ------------------ >*compose
          (S\NP)\NP
    ------------------ <apply
          S\NP
--------------- <apply
       S
```

Instead of allowing composition into the predicate via a slash-crossing version of forward composition, we must therefore assign the responsibility for permitting subject extraction to an extra lexical category for verbs like ***believe***. We will assume that such verbs, as well as being VP/S′ with an interpretation which we will write as BELIEVE′, bear the following special category, rather than VP/S:[14]

(55) believe := (VP/(S\NP))/NP: λx [BELIEVE′ (fx)]].

When we recall that the interpretation of a finite verb phrase S\NP over which the variable f ranges is a predicate, it is clear that this category merely transfers the functional composition illustrated in (53) into the lexical semantics of the verb.

The inclusion of this category allows derivations on the following

[14] This category is presumbly parallel to the 'raising to object' category (VP/(Sinf\NP))/NP of ***believe***, except that in order to ensure agreement between subjects and their verbs, we must ensure that the VP/(S\NP) that results from applying this function to a given NP restricts its argument appropriately for number, so that for example ***believe Harry*** is VP/(S\NPsng). I assume that the presence of the agreement feature prevents the passivisation of bare complement ***believe***. I am grateful to Jack Hoeksema and an anonymous referee for criticism of an alternative proposal in the draft.

pattern for leftward subject extraction, in which the complement predicate *eats apples* combines with *believe* by type-raising and backward crossing composition, like the adverbials in the last section:

```
(56)  Harry      I          can          believe          eats       apples
      --------- ---------- ----------- ---------------- ----------- ----
      St/(S/NP) S/(S\NP)   (S\NP)/VP   (VP/(S\NP))/NP   (S\NP)/NP   NP
                ----------------------->compose
                       S/VP
                                                        ------------------>apply
                                                              S\NP
                                                        ------------------<↑
                                                         VP\(VP/(S\NP))
                                       ------------------------------------<xcompose
                                                   VP/NP
                           --------------------------------------->compose
                                       S/NP
      ------------------------------------------------>apply
                           St
```

The analysis correctly allows phrases like *think won*$_{S/NP}$ (unlike simple predicates like *loves Mary*$_{S\backslash NP}$) to coordinate with transitive verbs by schema (17), giving rise to the following contrast:

(57)a. A man who(m) I like and think won.
b. *A man who(m) I like and loves Mary.

As far as rightward movement goes, the new category does not permit subjects to extract to the right out of simplex sentences, as in the ill-formed (58a) below, but it does allow subjects of complements to do so, as in the distinctly better (58b):

(58)a. *Was no good each Caesar whom Brutus ostensibly praised
b. ?Brutus implied was no good each Caesar whom he onstensibly praised.

Derivation (56) shows that a topicalised subject bears the same category St/(S/NP) as a topicalised object except that it must be marked for number, as St/(S/NPplr) on St(S/NPsng). A similar analysis applies for relatives: unboundedly extracted subject relative pronouns have to bear a category (N\N)/(S/NP), identical (except for agreement) to the object relative pronoun, and different from the subject relative pronoun with category (N\N)/(S\NP) found in simple subject relative clauses and illustrated in derivation (33).

Evidence for the rather surprising implication that there are *two* subject relative pronouns in English is provided by the existence of a fairly common dialect of English in which the subject relative pronoun in complex relative clauses is exactly the same lexical item *whom* as the object relative which it so closely resembles, giving rise to the following contrasts:

(59) a. a person whom I think loves Harry.

b. *a person whom loves Harry.

c. a person who loves Harry.

4. Multiple Dependencies and Functional Substitution

The important properties of sentence (1) and the related example (60) are: (a) that they have more than one dependency upon a single extracted Wh-item, and (b) that one of these gaps (indicated by subscript *p*) is in an island from which extraction would not normally be permitted, because it is a backward-applying function, not an argument.

(60) (articles) which I will file ___ without reading ___$_p$.

(61) a. (articles) which I will file ___ before reading your instructions.

b. *(articles) which I will read your instructions before filing ___.

Parasitic gaps are therefore unlike the multiple gaps which are permitted 'across the board' in the coordinate structures considered earlier.

4.1. *Parasitic Gaps in Adjuncts*

The lexical categories for example (60) are as follows:

(62)

(articles)	which	I will	file	without	reading
	(N\N)/(S/NP)	S/VP	VP/NP	(VP\VP)/VPing	VPing/NP
				(VP\VP)/NP (>*compose*)	

We can compose *without* and *reading*, but there the analysis blocks. Composition will not help, and nor will the coordination rule (since the categories of *file* and *without reading* are not the same). The introduction of some further operation or operations appears to be inevitable.

The intuition that sequences like *file without reading* constitute a semantically coherent entity of some kind of these sentences is very strong. The fact that such sequences can occur in isolation, in instructions like *shake before opening*, and that they can coordinate with transitive verbs in phrases like *file without reading and* $forget_{VP/NP}$ suggests that they are predicates of some kind – more specifically, that they bear the category of a transitive verb, VP/NP.[15] Szabolcsi (1983) proposed a combinatory rule to combine the VP/NP and the (VP\VP)/NP to yield this VP/NP. The rule was a special case of the following:

(63) $Y/Z \quad (X\backslash Y)/Z \;\Rightarrow\; X/Z.$

Such a 'substitution' rule, which is indexed <*xsubstitute* because the principal functor is on the right and the slashes are crossed in the same sense as in the backward crossing rule (16d), allows derivations like the following:

(64) (articles) which I will file without reading

which	I will	file	without	reading
(N\N)/(S/NP)	S/VP	VP/NP	(VP\VP)/VPing	VPing/NP

without reading ⟶ (VP\VP)/NP (>*compose*)

file + (VP\VP)/NP ⟶ VP/NP (<*xsubstitute*)

I will + VP/NP ⟶ S/NP (>*compose*)

which + S/NP ⟶ (N\N) (>*apply*)

C&G argues that the semantics of this rule corresponds to a third very basic combinator in Curry's system, called **S**. The definition of this combinator is given by the following equivalence:

(65) $\mathbf{S}FGx = Fx(Gx).$

It follows that the application of the combinator to two functions is equivalent to the following abstraction:

(66) $\mathbf{S}FG = [x]Fx(Gx).$

The substitution combinator abstracts over a variable in *both* the function and argument terms of an applicative expression, and all sentences that include parasitic gaps appear to reflect a syntactic combinator related to this operation. We therefore add the following combinatory rule parallel to the earlier functional application and composition rules (4) and (14):

(67) *Functional Substitution*:
A principal function over Z into functions-over-Y-into-X, of category $(X/Y)/Z$, $(X/Y)\backslash Z$, $(X\backslash Y)/Z$ or $(X\backslash Y)\backslash Z$ and interpretation F, may combine with an adjacent subsidiary

---

[15] This intuition is preserved in the GPSG analysis of parasitic gaps in Gazdar et al. (1985).

function into Y, of category Y/Z or $Y\backslash Z$ and interpretation G. The result is a function X/Z or $X\backslash Z$ from Z into X with interpretation $\mathbf{S}FG$.

The rule is subject to the usual principles (5) and (15) of Directional Consistency and Directional Inheritance. It will be convenient in discussing their effect to distinguish slashes according to the argument whose directionality they govern. In a function like $(X/Y)/Z$, I will refer to the slash which governs directionality with respect to the first argument Z as the "Z slash", and to the one governing directionality with respect to the second argument Y as the "Y slash".

Directional Consistency demands that the subsidiary function be on the side dictated by the Y slash. Directional Inheritance demands that all three Z slashes bear the same directionality. (Since the Z slash on the result must by definition (15) bear the same directionality as *all* corresponding Z slashes on the input side, when there is more than one input Z slash, they must necessarily be of the same directionality.) Therefore, as in the case of functional composition, the principles leave two degrees of freedom in specifying the directionality of the slashes in each instance of the rule, allowing the following four directional instances:

(68)a. $(X/Y)/Z{:}F \quad Y/Z{:}G \Rightarrow X/Z{:}\mathbf{S}FG$ (>*substitute*)
b. $(X/Y)\backslash Z{:}F \quad Y\backslash Z{:}G \Rightarrow X\backslash Z{:}\mathbf{S}FG$ (>*xsubstitute*)
c. $Y\backslash Z{:}G \quad (X\backslash Y)\backslash Z{:}F \Rightarrow X\backslash Z{:}\mathbf{S}FG$ (<*substitute*)
d. $Y/Z{:}G \quad (X\backslash Y)/Z{:}F \Rightarrow X/Z{:}\mathbf{S}FG$ (< *xsubstitute*)

Again, two of these are "forward' instances, with the principal functor on the left, and two are 'backward' instances, and again one of each is slash-crossing, while the other is non-slash-crossing. There are only two substitution rules that we should consider for English, the forward rule (68a) discussed in the next section, and the backward slash crossing rule (68d), which is the one required in the present example. These rules, should be like Backward Crossing Composition, restricted to apply only when Z is a major category, in order to prevent acceptance of sequences like *$good_{N/N}$ [*with a*]$_{(N\backslash N)/N}$ dog_N.[16]

Extraction of the first gap alone is allowed under the earlier analysis, of course, as in example (39) of section 3.1.1. But extraction from the second site alone, as in the corresponding example (36) in the same

[16] I am grateful to Jack Hoeksema for proposing this restriction.

section is not in general allowed. The new rule therefore not only allows the construction, but also captures the 'parasitic' nature of the second gap. At the same time it will not allow arbitrary double dependencies, such as

```
(69)   *(a man)  who(m)        I showed   — —P
                 (N\N)/(S/NP)  (S/NP)/NP
                 ------------------------ *
```

Examples like the following provide further evidence for the mechanism for extracting inner arguments from section 3.1.2 and for the existence in the grammar of constituents like *put on the table*$_{VP/NP}$.

```
(70)  ...  which         I will  put on the table  without            reading
           (N\N)/(S/NP)  S/VP    VP/NP             (VP\VP)/VPing      VPing/NP
                                                   ------------------------------ >compose
                                                           (VP\VP)/NP
                                 ---------------------------------------------- <xsubstitute
                                                 VP/NP
                         ------------------------------------- >compose
                                       S/NP
           ------------------------------------ >apply
                      (N\N)
```

4.2. *Multiple Dependencies in Arguments*

Both the island character of adverbials like *without reading Proust*, and the parasitic character of extractions from them, arise from their backward combining category VP\VP. However, there is a second variety of multiple unbounded dependency in which both gaps are in *arguments* of a verb. These constructions, of which the following sentence is an example, are noticeably less acceptable than the ones in the previous section, but could be allowed by including a further instance from the four substitution rules which are permitted by the Principles of Consistency and Inheritance, the Forward, non-crossing version (68a).[17]

[17] The category (VP/(Si\NP))/NP given here for the object control verb *persuade* is assumed without further comment, but it is consistent with extraction possibilities under the present model, and with the possibilities for reflexivisation and passivisation, under the assumption that the latter rule applies to first NP arguments.

(71)

```
?(men)   who(m)     I will      persuade         every friend of   to vote for
       (N\N)/(S/NP)  S/VP   (VP/(Si\NP))/NP         NP/NP .       (Si\NP)/NP
                            ----------------------------- >compose
                                 (VP/(Si\NP))/NP
                                 -------------------------------------------- >substitute
                                                  VP/NP
                     ---------------------------------------- >compose
                                     S/NP
       ------------------------------------------------ >apply
                     (N\N)
```

The present theory allows *either* extraction site in these examples to be accessible to simple extraction, so that neither exhibits the parastic behavior of the extractions from adverbial postmodifiers of the previous section. The reason is that the theory distinguishes between adjunct functions and subcategorised arguments. The forward substitution rule would therefore permit the following pattern:

(72) a. a man whom I will persuade every friend of to vote for

b. a man whom I will persuade every friend of my mother to vote for

c. a man whom I will persuade every friend of to vote for my mother.

However, many examples which this rule would permit are less acceptable. The doubly gapped construction below seems to be rather borderline:

(73)

```
*?(a man)    whom      I will  send a picture of    to
          (N\N)/(S/NP)  S/VP     (VP/PP)/NP       PP/NP
                                 ------------------------ >substitute
                                          VP/NP
                         --------------------------- >compose
                                     S/NP
          ------------------------------------- >apply
                     (N\N)
```

(It is tolerated by Engdahl (1983), and by Chomsky (1982), but Chomsky (1981) rejects it, as do Sag (1983) and Contreras (1984)). Many sentences which the rule potentially allows are much worse – for example:

(74) *(a man) whom I will show to

```
(N\N)/(S/NP)  S/VP  (VP/PP)/NP  PP/NP
                    ----------------->substitute
                          VP/NP
              -------------------->compose
                       S/NP
----------------------------->apply
          (N\N)
```

The following would also be permitted, under the assumptions of this paper:

(75)a. ?Which man did you persuade to believe that you like

b. *Which man did you say believes that you like

– although the rule still will not allow (69), **a man whom I showed* $t\,t_p$. Engdahl (1983, p. 24) has suggested that the possibility of parasitic gapping shows a striking parallel to the constraints on possible coreference of pronouns and bound anaphors, and Sag (1983) has suggested capturing such a constraint at the level of interpretation, as have Gazdar et al. (1985). The present theory does not appear to offer any new insight on this question.

4.3. *Subject Parasitic Gaps*

Many apparent opportunities for parasitic gap constructions with subject gaps are not allowed. Engdahl (1983, ex. 54–56) gives examples related to the following:

(76)a. *(a man) who __ painted a picture of __$_p$

b. *(a man) who __ remembered talking to __$_p$

c. *(a man) who __ remembered that John talked to __$_p$.

As she points out, the illegality of these sentences does not arise from an overall ban on subject gaps in parasitic constructions, or from a requirement that the gaps have parallel grammatical functions, as the following well-formed examples show:

(77)a. (the Caesar) whom Brutus will imply __ was no good whilst ostensibly praising __$_p$

b. (a man) who every boy who meets __$_p$ admires __

c. (a man) who you said John's criticism of __$_p$ would make us think __ was stupid.

The first example has a non-subject gap parasitic on an embedded subject gap. The second has a non-subject gap *inside* a subject which appears to be parasitic upon a non-subject gap, while the third has one which appears to be parasitic on an embedded subject gap.

The examples (76) are not permitted by the substitution rule, because the two functors which contain the respective gaps and which can be assembled by the forward composition rule are not of the appropriate form for any instance of functional substitution to apply:

(78)a. *(a man) who [painted]$_{(S\backslash NP)/NP}$ [a picture of]$_{(NP/NP)}$

b. *(a man) who [remembered]$_{(S\backslash NP)/VPing}$ [talking to]$_{VPing/NP}$

c. *(a man) who [remembered]$_{(S\backslash NP)/S'}$ [that John talked to]$_{S'/NP}$.

On the other hand, (77a) is allowed, because the subject-extracted function *imply was no good*, built analogously to the phrase *believe eats apples* in example (56) by type-raising the VP *was no good*$_{S\backslash NP}$ and backward-composing it with the verb *imply*$_{(VP/(S\backslash NP))/NP}$, and the adjunct function *while ostensibly praising*, built by the forward composition rule, are of the appropriate form and linear order for the backward instance of the substitution rule to apply, thus:[18]

(79)

```
... who(m)    [Brutus will] [imply was no good] [while ostensibly praising]
(N\N)/(S/NP)   S/VP          VP/NP               (VP\VP)/NP
                             ------------------------------------ <xsubstitute
                                          VP/NP
               ------------------------------------ >compose
                             S/NP
-------------------------------------- >apply
               (N\N)
```

A related ill-formed sentence in which the subject gap is not embedded is, like other examples relating to nominative island constraints, prevented by the directionality of the argument of the predicate category S\NP. (The omnivorous category that we are abbreviating to VP\VP is written S\S here to make it otherwise potentially combinable with S.)

[18] The constituent *imply was no good* really bears a category VP/NPsng, distinguished for the number of the extracted subject, while the non-subject argument of the constituent *while ostensibly praising* is not distinguished in this way. The substitution rule must therefore ensure that the argument Z of the function that it produces bears the union of the feature restrctions of the argument Z in its input functions.

(80) ... *who was absolutely useless despite Brutus extravagantly praising
(N\N)/(S/NP) S\NP (S\S)/NP
——————————— *

It is important to note that the impossibility of this combination is not language specific. The principle of Directional Inheritance means that not only is there no instance in English of substitution that will equate the arguments of a forward combining and a backward combining function, but that *no* such instances are permitted by the principles, because the NP arguments have different directionality.

A similar asymmetry holds for parasitic subject gaps in phrases like *burn t without realising t_p was valuable*. These would be allowed on the assumption that the embedded subject gap gives rise to (VP\VP)/NP. But a non-embedded subject gap in the adjunct is impossible, not only because it would require slash-crossing forward composition in English, but because the Principle of Inheritance means that no instance of substitution that could combine the result is permitted:

(81) *(who(m) did you) meet before left
VP/NP (VP\VP)/S S\NP
——————— *
(VP\VP)\NP
——————————— *

Because the second type of multiple dependencies, discussed in the last section, are also mediated by a (forward) instance of the substitution rule, the theory makes a parallel prediction concerning extracted subjects in that construction. Embedded subjects should be able to take part in multiple dependencies, but the subject itself should not. The following example related to Chomsky's (1986, example 123b), involving an embedded subject and the forward substitution combinator, is parallel to (79), and seems fairly acceptable:

(82) A man who, [although Harry admired,]$_{(S/S)/NP}$ [Brutus said was no good]$_{S/NP}$.

In particular it is comparable in acceptability to the corresponding across the board example *a man who [Harry praised and Brutus implied was no good]*$_{S/NP}$, and *a man who, although Harry admired, [I disliked]*$_{S/NP}$.

On the other hand, the following example parallel to (80) does not seem to be as acceptable:

(83) *A man who, [although Harry admired,]$_{(S/S)/NP}$ [was in fact no good]$_{S\backslash NP}$.

This example is ruled out, according to the present theory, because *no* syntactic combinatory rule is allowed to equate two arguments of different directionality. The example is parallel to Chomsky's (1986, example 122b) (which I would also star):

(84) A man who [whenever I meet]$_{(S/S)/NP}$ [looks old]$_{S\backslash NP}$.

Following Longobardi, he notes that the example is better than *a man who looks old whenever I meet*, which it undoubtedly is. However, it is still a very bad sentence. In particular, it seems worse than the corresponding object construction *A man who, whenever I meet, I insult*, and no better than the corresponding across the board example **a man who I met and looks old*, which is generally regarded as ungrammatical. If these judgements in this rather grey area are sound, then Engdahl's (1984) observation that an 'empty subject' of tensed S cannot license a parasitic gap, except when it is embedded in a bare complement, is correct. It emerges in the present theory as a direct consequence of the directionality of the subject in an SVO language, and the constraints embodied in the Principle of Inheritance.

Of the remaining cases of Engdahl's subject parasitics, example (77b), repeated here, is slightly different:

(85) (a man) whom every boy who meets $\_\__p$ admires __.

This time it is the first gap which appears to be the parasitic member of the pair, or so the ill formedness of (86b) below in comparison with (86a) suggests:

(86)a. (a man) whom everyone who likes politicians admires.

b. *(a man) whom everyone who likes admires politicians.

However, example (86b) is really a violation of the Subject Condition of Chomsky (1970). Whatever the basis of this constraint, it applies in present terms to backward slash-crossing composition into subject NPs. According to the present theory, it is not surprising that subjects should be able to take part in *multiple* dependencies, since those arise from a different rule. Thus the backward (crossing) instance (68d) of the substitution rule could combine *admires*$_{(S\backslash NP)/NP}$ with *every boy who meets*$_{NP/NP}$, to accept (85) as follows:

(87)

```
(a man)  who(m)         every  boy       who            meets      admires
         (N\N)/(S/NP)   NP/N   N/(N\N)   (N\N)/(S\NP)   (S\NP)/NP  (S\NP)/NP
                        ------------->compose
                          NP/(N\N)
                          ------------------------>compose
                                  NP/(S\NP)
                                  ---------------------------->compose
                                              NP/NP
                                              ---------------------------<xsubstitute
                                                         S/NP
         -------------------------------------------------------->apply
                                  (N\N)
```

(This derivation assumes that quantifiers like *every* license their complement nouns to type raise over adjuncts, and thereby allow extraction from Wh-islands.)

3. Conclusions: Combinators and Universal Grammar

The concepts of functional application and abstraction are so general as to make it inevitable that natural language grammars should look like *some* kind of applicative system. And by committing ourselves to the Principle (3) of adjacency, we in effect commit ourselves to the use of combinators. But the particular system to which the linguistic data of sections 1 and 2 has led us poses two specific questions. The first is: why should natural languages adhere to the principle of Adjacency anyway? Why should they be explicitly combinatory *at all*? Why shouldn't they take the form of some other kind of applicative system, such as the lambda calculus itself? The second is: why do they use *these* combinators, and not some others, such as Curry's minimal **S-K** system?

The whole point of the combinators is that they allow the definition of languages up to the full expressive power of the Lambda calculus – that is, languages in which functions can be defined in terms of other functions – without the use of abstraction and bound variables. One reason for the involvement of combinators in natural grammars could therefore be that there is a pressure to minimise the use of bound variables.

The companion to the present paper, C&G, points out that the use of variables, via "environments" embodied in association lists or other data-structures, is a major source of computational overheads in practical programming languages, and that Turner (1979a, 1979b; see also Burge 1975; Buneman et al. 1982; Statman, 1986) has argued that the applicative or functional programming languages (that is, languages closely related to the lambda calculus, lacking assignment and side-effects), can be efficiently evaluated by first compiling them into equivalent expressions using combinators to exclude bound variables, and then evaluating the combinatory machine code by purely substitutive methods. There are strong similarities between the set of combinators proposed by Turner and the system proposed here for natural language.

To claim that natural language semantics keeps the use of bound variables to a minimum is not to claim that natural languages eschew bound variables altogether. There certainly exist entities in natural language, such as 'bound variable' and 'resumptive' pronouns, which seem to be just like bound variables. Natural language semantics appears to be a mixed system, in which certain frequently required abstractions – most obviously, the one corresponding to functional composition – are accomplished without bound variables. However the computationally expensive (but completely general) alternative using bound variable pronouns is reserved for more complex cases. This proposal seems to be borne out by the fact that in languages which permit Wh-movement with and without resumptive pronouns – for example, Irish (McCloskey, 1978) – the two varieties appear to be in complementary distribution, rather than free variation, with the resumptive strategy reserved for extraction sites which are low on the accessibility hierarchy of Keenan and Comrie (1977). Moreover, resumptive and bound variable anaphora is notoriously free of the constraints that limit Wh-movement.

There is a second reason for the grammar of natural languages to include rules based on **S**, **B**, **C**$_*$, and the like, rather than on combinators like Schönfinkel's **K**. All of the combinators used here are PROCEDURALLY NEUTRAL in the sense used by Pareschi and Steedman (1987): specifying their result determines (not necessarily uniquely) the arguments they applied to. It follows that they conform to the combinatory equivalent of a recoverability condition (Katz and Postal 1964). Combinators like **K** are not invertible and therefore induce the combinatory equivalent of unrecoverable deletion.

According to the present theory, a wide range of puzzling constructions in natural language syntax that have been described in terms of unbounded movement, including the parasitic gapping constructions, receive a simple and unified explanation in terms of a few very basic combinators of the type used in the foundations of logic and the lambda calculus. The particular combinators that are implicated are those that conform to a 'recoverability condition'. The possible realisations of those combinators as related syntactic operations are restricted by the principles of Consistency and Inheritance, which are claimed to be universal, and to be consequences of the semantics of the theory itself, rather than stipulations. In such a theory, many phenomena of unbounded dependency emerge as consequences of function-argument relations, intuitively the natural foundation for notions like 'government', and 'head of a construction', whether in their traditional form of their more recent incarnations. An applicative system including these combinatory rules accounts for the relevant constructions without invoking such additional independent graph-theoretic notions of Government-Binding as "projection path", and "correctedness", and without invisible "empty categories" and attendant principles and conditions. While the present theory is more closely related to GPSG (and even more so the Head Phrase Structure Grammars of Pollard, 1988, it does not treat "slash" as a feature subject to attendant conventions and feature coocurrence restrictions. Instead, extraction phenomena are directly projected from the function-argument relations of the lexicon by the combinatory rules. Many phenomena which have given rise to "constraints on movement" and the "empty category principle" are simply emergent consequences of the nature of these rules.

The implication is that the form of natural language syntax may be determined by the fact that its semantics is expressed in combinatory form, together with a requirement to maintain a very transparent type driven relation between the two. (Such a need would be explicable on grounds of easy learnability, and on other grounds of economy). The way in which the combinators minimise the use of bound variables, together with the efficiency which variable-free systems induce for certain types of computation, make it seem plausible that the semantics in turn takes this form because of the practical advantages that accrue to an applicative system expressed in combinatory form.

References

Ades, Anthony E. and Mark J. Steedman: 1982, 'On the order of words', *Linguistics and Philosophy* **4**, 517–58.

Ajdukiewicz, Kazimierz: 1935, 'Die syntaktische Konnexität', *Studia Philosophica* **1**, 1–27. English translation in Storrs McCall (ed.), *Polish Logic 1920–1939*, Oxford University Press, pp. 207–31.

Bach, Emmon: 1979, 'Control in Montague Grammar', *Linguistic Inquiry* **10**, 515–31.

——: 1980, 'In Defense of Passive', *Linguistics and Philosophy* **3**, 297–341.

——: 1983, 'Generalised Categorial Grammars and the English Auxiliary', in Frank Heny and Barry Richards (eds.), *Linguistic Categories: Auxiliaries and Related Puzzles, II*, D. Reidel, Dordrecht, pp. 101–20.

Bar-Hillel, Yehoshua: 1953, 'A Quasi-Arithmetical Notation for Syntactic Description', *Language* **29**, 47–58.

Bresnan, Joan W.: 1972, *Theory of Complementation in English Syntax*, unpublished Ph.D. dissertation, MIT, Cambridge, Massachusetts.

Buneman, Peter, Robert E. Frankel, and Rishiyur Nikhil: 1982, 'An Implementation Technique for Database Query Languages', *ACM Transactions on Database Systems* **7**, 164–86.

Burge, William H.: 1975, *Recursive Programming Techniques*, Addison-Wesley, Reading, Massachusetts.

Chomsky, Noam: 1970, 'Remarks on Nominalisation', in R. Jacobs and P. Rosenbaum (eds.), *Readings in English Transformational Grammar*, Ginn, Waltham, Massachusetts, pp. 184–221.

——: 1981, *Lectures on Government and Binding*, Foris, Dordrecht.

——: 1982, *Some Concepts and Consequences of the Theory of Government and Binding*, MIT Press, Cambridge, Massachusetts.

——: 1986, *Barriers*, MIT Press, Cambridge, Massachusetts.

Chomsky, Noam and Howard Lasnik: 1977, 'Filters and Control', *Linguistic Inquiry* **8**, 425–504.

Contreras, Heles: 1984, 'A Note on Parasitic Gaps', *Linguistic Inquiry* **15**, 698–700.

Curry, Haskell B.: 1961, 'Some Logical Aspects of Grammatical Structure', *Structure of Language and its Mathematical Aspects. Proceedings of the Symposia in Applied Mathematics* **XII**, American Mathematical Society, Providence, Rhode Island, pp. 56–68.

Curry, Haskell B. and Robert Feys: 1958, *Combinatory Logic: Vol. I*, North-Holland, Amsterdam.

Dowty, David: 1982, 'Grammatical Relations and Montague Grammar', in Pauling Jacobson and Geoffrey K. Pullum (eds.), *The Nature of Syntactic Representation*, D. Reidel, Dordrecht, pp. 79–130.

Dowty, David: 1988, 'Type Raising, Functional Composition, and Non-Constituent Conjunction', in Richard T. Oehrle, E. Bach and D. Wheeler (eds.), *Categorial Grammars and Natural Language Structures*, D. Reidel, Dordrecht, (in press).
Language Structures, D. Reidel, Dordrecht, (in press).

Engdahl, Elisabet: 1981, 'Multiple Gaps in English and Swedish', in Torsten Fretheim and Lars Hellan (eds.), *Proceedings of the 6th Scandanavian Conference of Linguistics*, Tapir, Trondheim.

Engdahl, Elisabet: 1983, 'Parasitic Gaps', *Linguistics and Philosophy* **6**, 5–34.

Engdahl, Elisabet: 1984, 'Parasitic Gaps, Resumptive Pronouns, and Subject Extractions', manuscript, University of Wisconsin, Madison.

Gazdar, Gerald: 1981, 'Unbounded Dependencies and Coordinate Structure', *Linguistic Inquiry* **12**, 155–84.

Gazdar, Gerald: 1982, 'Phrase Structure Grammar', in Pauline Jacobson and Geoffrey K. Pullum (eds.), *On the Nature of Syntactic Representation*, D. Reidel, Dordrecht, pp. 131–86.

Gazdar, Gerald, Ewan Klein, Ivan A. Sag, and Geoffrey K. Pullum: 1985, *Generalised Phrase Structure Grammar*, Blackwell, Oxford.

Geach, Paul T.: 1972, 'A Program for Syntax', in Donald Davidson and Gilbert Harman (eds.), *Semantics of Natural Language*, D. Reidel, Dordrecht, pp. 483–97.

Huck, Geoffrey: 1985, *Discontinuity and Word-order in Categorial Grammar*, unpublished Ph.D. dissertation, University of Chicago.

Joshi, Aravind: 1987, 'The Convergence of Mildly Context-Sensitive Formalisms', Paper to CSLI Workshop on Processing of Linguistic Structure, Santa Cruz, Jan 1987, manuscript, University of Pennsylvania.

Katz, Jerrold and Paul Postal: 1964, *An Integrated Theory of Linguistic Descriptions*, MIT Press, Cambridge, Massachusetts.

Keenan, Edward and Bernard Comrie: 1977, 'Noun Phrase Accessibility and Universal Grammar', *Linguistic Inquiry* **8**, 63–100.

Klein, Ewan and Ivan A. Sag: 1984, 'Type-Driven Translation', *Linguistics and Philosophy* **8**, 163–201.

Kuno, Susumo: 1973, 'Constraints on Internal Clauses and Sentential Subjects', *Linguistic Inquiry* **4**, 363–86.

Lambek, Joachim: 1958, 'The Mathematics of Sentence Structure', *American Mathematical Monthly* **65**, 154–70.

——: 1961, 'On the Calculus of Syntactic Types', *Structure of Language and its Mathematical Aspects. Proceedings of the Symposia in Applied Mathematics*, **XII**, American Mathematical Society, Providence, Rhode Island, pp. 166–78.

Lyons, John: 1968, *Introduction to Theoretical Linguistics*, Cambridge University Press.

McCloskey, M. James: 1978, *A Fragment of a Grammar of Modern Irish*, Ph.D thesis, University of Texas at Austin, *Texas Linguistic Forum* **12**.

Moortgat, Michael: 1988, 'Mixed Composition and Discontinuous Dependencies', in Richard T. Oehrle, E. Bach, and D. Wheeler (eds.), *Categorial Grammars and Natural Language Structures*, D. Reidel, Dordrecht, (in press).

Morrill, Glyn: 1987, *Meta-categorial Grammar*, Ph.D. thesis, University of Edinburgh (in preparation).

Oehrle, Richard T.: 1988, 'Multidimensional Compositional Functions as a Basis for Grammatical Analysis', in Richard T. Oehrle, E. Bach and D. Wheeler (eds.), *Categorial Grammars and Natural Language Structures*, D. Reidel, Dordrecht (in press).

Pareschi, Remo: 1985, 'Combinatory Categorial Grammar, Logic Programming, and the Parsing of Natural Language', DAI Working Paper, University of Edinburgh.

Pareschi, Remo and Mark Steedman: 1987, 'A Lazy Way to Chart Parse with Categorial Grammars', paper to ACL Conference, Stanford, July 1987, manuscript, CIS, University of Pennsylvania.

Perlmutter, David M.: 1971, *Deep and Surface Structure Constraints in Syntax*, Holt Rinehart and Winston, New York.

Pollard, Carl J.:1988, 'Categorial Grammar and Phrase-Structure Grammar: An Excursion on the Syntax-Semantics Frontier', in Richard T. Oehrle, E. Bach, and D. Wheeler (eds.), *Categorial Grammars and Natural Language Structures*, D. Reidel, Dordrecht, (in press).

Ross, John Robert: 1967, *Constraints on Variables in Syntax*, unpublished doctoral dissertation, MIT, Cambridge, Massachusetts.

Sag, Ivan A.: 1983, 'On Parasitic Gaps,' *Linguistics and Philosophy* **6**, 35–45.

Shieber, Stuart: 1986, *An Introduction to Unification-Based Approaches to Grammar*, CSLI, University of Stanford, California.

Schönfinkel, Moses: 1924, 'Über die Bausteine der Mathematischen Logik', *Mathematische Annalen* **92**, 305–16.

Statman, Rick: 1986, 'On Translating Lambda Terms into Combinators', *Proceedings of the Symposium on Logic in Computer Science*, Cambridge, Massachusetts, June 1986, pp. 378–82.

Steedman, Mark J.: 1985, 'Dependency and Coordination in the Grammar of Dutch and English', *Language* **61**, 523–68.

——: 1987 'Gapping Constituent Conjunction', manuscript, University of Pennsylvania.

Steedman, Mark J. and Anna Szabolcsi: 1987, 'Against Empty Categories', manuscript, Hungarian Academy of Science.

——: 1988, 'Combinators and Grammars', in Richard T. Oehrle, E. Bach, and D. Wheeler (eds.), *Categorial Grammars and Natural Language Structures*, D. Reidel, Dordrecht (in press).

Szabolcsi, Anna: 1983, 'ECP in Categorial Grammar', manuscript, Max Planck Institute, Nijmegen.

——: 1985, 'Combinators and the Projection Principle', paper to the Conference on Categorial Grammar, Tucson, Arizona, June 1985.
——: 1987, 'Filters vs. Combinators', in Bodnár, Máté and Pólos, *Intensional Logic, Semantics and the Philosophy of Science: in Honour of Imre Ruzsa*, FTKK, Budapest.
Taraldsen, K. Tarald: 1979, 'The Theoretical Interpretation of a Class of Marked Extractions', in A. Belletti, L. Brandi, and L. Rizzi (eds.), *Theory of Markedness in Generative Grammar*, Scuole Noarmale Superiore di Pisa, pp. 475–516.
Turner, David A.: 1979a, 'A New Implementation Technique for Applicative Languages', *Software – Practice and Experience* **9**, 31–49.
——: 1979b, 'Another Algorithm for Bracket Abstraction', *Journal of Symbolic Logic* **44**, 267–70.
Wall, Robert E.: 1972, *Introduction to Mathematical Linguistics*, Prentice Hall, Englewood Cliffs, New Jersey.

Received 1 April 1986
Revised 4 March 1987

Centre for Cognitive Science and Department
of Artificial Intelligence
University of Edinburgh
Edinburgh, U.K. EH8 9LW

and

Dept. of Computer & Information Science
University of Pennsylvania
Philadelphia, PA 19104-6389
U.S.A.