Chapter 12

On ''The Computation'' *(Draft 2.0, March 13, 2006)*

Mark Steedman

12.1 Introduction

The Minimalist Program proposed by Chomsky (1993, 1995a,b, 2001) identifies the language system as a mapping or "computation" between two "interface levels" of phonetic or phonological form (PF) and logical form (LF). The interface levels correspond intuitively to representations of sound and meaning, and act as interfaces with the systems for perceiving and realizing auditory and visual forms of language on the one hand, and for establishing truth and inferential consequence on the other. The term "computation" is used in approximately the sense of Marr (1977), to denote the "theory of the computation" from one level to the other, as distinct from any one of a number of algorithms that might realize the computation. (Thus "computation" here means approximately the same as Chomsky's earlier term "competence", as opposed to "performance".)

These assumptions are very natural and appealing, and were embraced by a number of earlier base-generative or monostratal theories of grammar. However, a number of features of the Minimalist Program make it more complex than this simple statement might suggest. The interface level of PF incorporates some quite complex processes of "remerging" or deletion under identity which seem to go beyond the transductions that one would normally refer to as an interface. Similarly, operations of "covert" movement such as quantifier raising intervene between the representation that determines PF and that which determines LF. The present paper seeks to simplify this picture by refining the theory of the intervening Computation, drawing on work in other frameworks.

Questions concerning the nature of the computation come in two general forms, both first posed in early work by Chomsky (1957, 1959). The first concerns the automata-theoretic and expressive power of the system, which Chomsky convincingly argued to be greater than that of Context-Free Grammar (CFG) and the associated class of Push-Down Automata (PDA). While mainstream linguistics subsequently appeared to lose interest in the question of *how much* more expressive power was needed for linguistically realistic grammars, subsequent work in computational linguistic frameworks strongly suggests that it may be only minimally greater. The second kind of question concerns empirical generalizations about universal properties of attested natural grammars, of which mainstream linguistics has been immensely productive.

The answer to questions of these two kinds has been the principal preoccupation for the past forty or fifty years of the two main formal approaches to the theory of grammar, the computational and the generative. Recently, these traditions have shown signs of converging. In what follows, it will become clear that many of the key elements of a modern theory of grammar are distinctively computational in origin.

12.2 Theory of Grammar: the State of the Art

The main problem for any theory of grammar is posed by the presence at the surface or PF level of numerous semantically discontinous constituencies and fragmentary non-constituents, arising in constructions like relative clauses and coordinate structures, as well as more abstruse constructions such as parentheticals and intonational phrasing (Chomsky 1975:210-211, section written c.1956). For example, the following are some simple cases in which verb-complement relations are discontinuous and/or various nonconstituent fragments appear to behave like grammatical constituents for purposes of coordination:

(1) a. The theory that Monboddo proposed.
    b. Monboddo proposed, and Johnson ridiculed, the theory of monogenesis.
    c. Monboddo gave Johnson a book and Boswell a pamphlet.

Such phenomena have given rise to the broad range of theories known as Transformational Grammar (TG) in which the mapping from PF to LF is mediated by one or more intermediate levels of structural representation, related to the interface levels by processes of movement and/or deletion (or its inverse, copying).

Rule systems allowing movement and deletion were shown quite early on to be of essentially unlimited expressive power (Peters and Ritchie 1973). Both for reasons of theoretical parsimony and for reasons of computational complexity, this result gave rise to a search within both generative and computational frameworks for more constrained grammar formalisms.

A key insight behind this development was Kimball's 1967 and Emonds' 1970; 1976 observation that most movement rules were "structure preserving"—that is, that they moved elements to positions (such as the subject position) that were independently specified in the base grammar. The significance of this observation (and the related proposal by Woods 1970 to achieve the effect of certain transformations computationally using a fixed set of "registers" corresponding to subject, object, etc.) was that it offered a way to make movement "monotonic", in the sense of not destroying or modifying structure once built. This observation, among others, led to a number of proposals to "base generate" such contructions—that is, to specify them directly in the context-free base grammar or equivalent, rather than deriving them by subsequent structural change—including Kuno 1965; Thorne, Bratley and Dewar 1968; Woods 1970; and Brame 1978. Since the majority of these constructions were *bounded*—that is, defined by movements confined to the domain of a single tensed clause—there were also a number of proposals to handle base generation *lexically*, by associating a full specification of the local domain with the heads of constructions, and in particular with verbs, notably by Oehrle (1975), Dowty (1978) and Bresnan (1978). There were also proposals to handle the *unbounded* constructions—relativization and its kin, together with certain related coordination phenomena, via base generation, including Thorne, Bratley and Dewar 1968; Woods 1973; Joshi, Levy and Takahashi 1975; Koster 1978; Gazdar 1981; Ades and Steedman 1982; Steedman 1985; and Gazdar et al. 1985.

Many of these developments were rapidly assimilated by mainstream generative grammar. For example, in the "Government Binding" (GB, see Chomsky 1981) version of TG that was standard from the mid 1970s through the 1980s, the mapping between PF and LF was mediated by a representational level of "S-structure," itself specified generatively via a base grammar defining an underlying level of "D-structure" and a general rule of (overt) movement hedged around by constraints on legal output S-structures (such as the Binding Conditions). S-structure was then mapped to PF (sometimes still referred to as Surface Structure) by processes of deletion, and to LF by processes of (covert) quantifier movement. While this account preserved the derivational character of earlier versions of TG by involving movement in the generation of S-structure, and some argumentation was offered for the theoretical necessity of an autonomous level of D-structure, the very generality of the movement rule, and the fact that the essential constraints upon it were defined in terms of its structural inputs, mean that this theory was essentially monotonic, and essentially base generative in all but name.

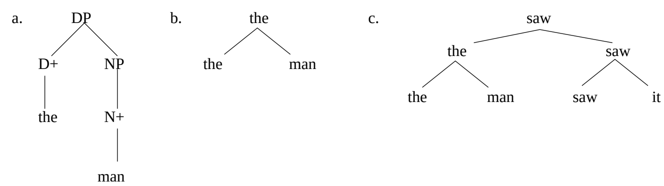
More recently, as the other chapters in this volume attest, the Minimalist Program has sought to modify this theory by the elimination of S-structure (and also D-structure) as autonomous structural levels of representation. The resulting theory in its present form, as presented in those chapters and elsewhere, seems to amount to the following.

First, there is a single underlying level of representation, including traces, corresponding to some form of the erstwhile S-structure, built in a single derivational process or "computation", combining operations of structural merger and movement, accompanied by (possibly "multiple") events of language-specific "spell out" of the elements of PF.

The details of this process are as follows.[1]

1. The sentence-level representation is derived from a "Numeration" or unordered multiset of lexical items, say "it", "saw", "the", and "man". These lexical items are selected at random and merged to build structure bottom-up in the process referred to as the Computation.

2. The merging operation projects head categorial information from whichever of the merged items is the head, with very little of the unary branching substructure that we have become used to from the $\bar{X}$-theory of the 1970s. Thus in place of trees like (a) we get trees like (b) and (c) (Chomsky 1995a,b:246-247).



(2)

3. A further notion of "phase" is defined over such structures (Chomsky

2001; Svenonius 2004). The domains defined as phases correspond to the notion of (extended) projection domain of a head, such as *the* or *saw*, and are strongly reminiscent of those defined as “kernel sentences” in Chomsky 1957, and the elementary trees of Tree-Adjoining Grammar (TAG, Joshi, Levy and Takahashi 1975), the Lexical-Functional specifications of LFG (Bresnan 1982), and the category domains of Combinatory Categorial Grammars (CCG, see below).

4. The processes of “merging” phases resemble the application of “generalized” or “double-based” transformations of Chomsky 1957, as they do the twin processes of substitution and adjunction of TAG, the unification apparatus of LFG, and the various combinatory reductions of CCG (although, as we shall see below, the latter differ in being purely type-driven, rather than structure-dependent as transformational and tree-adjoining rules quintessentially are.)

5. From time to time during the derivation or computation, various conditions may trigger a movement. The structuring of the developing representation into phases corresponding to the local projections of nominal and verbal heads makes these movements successive-cyclic. Move is in 1957 terms a singulary transformation: the base generative theories by definition have no corresponding surface syntactic operation, (although they must of course express the underlying LF relation in the semantics of base generation). Further processes of identifying and labeling identical substructures relevant to their subsequent deletion at the level of PF (the labeling possibly accomplished by re-merging or equating the relevant elements of the numeration) is also defined as part of this process (Chomsky 1995b:252-254).

6. More recently, this latter work has been subsumed under the mechanism of multiple spell-out, as one might expect (at least in the bounded cases) from the general kinship of multiple spell-out and phase to lexicalization in the strong sense of theories like HPSG, whose notion of structure-sharing is taken over for the purpose. In this version, PF is defined without further computation. However, some important grammatical operations—notably deletion under coordination—take place at PF post-spell-out—see Chomsky 1995b:202-205.) Any further move-

ments (such as quantifier movement) have an effect on the logical form, but are "covert"—that is, they do not affect PF.

The involvement of movement preserves the "derivational" character of the theory, but as in the earlier incarnation, the structure-preserving nature of the movement means that its effects are essentially monotonic, and limited in a very similar way to base generation.

However, there are a number of surprising features to this theory which set it apart from the base-generative relatives mentioned earlier. One arises from the involvement of the Numeration. If one regards Minimalism as a program for a generative theory of grammar—that is, one which will nondeterministically enumerate the sentences of the languages it applies to—then it seems as though we must begin by enumerating the set of all possible Numerations, applying the computational system as a filter to obtain the corresponding set of strings of the language, as seems to be proposed by Chomsky 1995b:227,237. The set of strings corresponding to a random numeration will usually be empty (as in the case of the numeration {"to","to", "to"}), but will otherwise usually include several sentences, including some with unrelated meanings. (For example, the numeration {"loves", "John", "Mary"} will support one family of sentences in which the subject is John, and another in which it is Mary.)

Of course, there is nothing technically wrong with setting up the generative system this way. But if we are going to do *that*, why not go the whole hog and use the set of strings over the lexicon of the language as the set of Numerations, rather than multisets??

One reason is that the structures that are generated by the computational system on the way to LF are by assumption not linearly ordered (Chomsky 1995b:334). Language-specific order, must therefore be imposed post-Spell-out, by reference to parameter settings of the language, such as head finality, possibly with the aid of some fine tuning via "economy principles," or "optimality conditions", or via general principles relating command at LF to linear order, as with Kayne's 1994 Linear Correspondance Axiom. This is a feature shared with some other grammar formalisms that separate "immediate dominance" (ID) and "linear precedence" (LP) rules. However, in the case of Minimalist grammars it seems to be quite hard to decide whether these very general principles have been successfully applied to capture all and only the sentences of a significant fragment of a specific language without over- or under- generation, or how the number of degrees of freedom required to do so compares with those exploited in other frameworks. A key observation that has been made in other frameworks is that a) such principles are usually confined to defining order among sister nodes, and b) unless they are so confined, they are of very high expressive power indeed (Ojeda 1988). This in turn suggests that such principles can and should be lexicalized.

It seems worth asking if there is a way to make Minimalism a little more minimalist by drawing on proposals from within the base-generative and computational traditions, in order to specify the Computation in a way that will simplify the interface levels by clarifying the role of the Numeration and its relation to the process of multiple spell-out and phase (all of which one might expect to be related to the process of lexical insertion), bringing word-order under lexical control, and conflating Move with Merge, as has been proposed within the categorial tradition by Bach (1976), Dowty (1978), and Ades and Steedman (1982), and in the transformationalist tradition by Koster (1978), Berwick and Epstein (1995), and Epstein et al. (1998).

12.3 Combinatory Categorial Grammar

Combinatory Categorial Grammar (CCG, Steedman 2000b, hereafter, *SP*), like all other varieties of categorial grammar (Ajdukiewicz 1935, Bar-Hillel 1953, Bach 1976; Dowty 1978 Oehrle, Bach and Wheeler 1988; Buszkowski, Marciszewski and van Benthem 1988; Wood 1993) is a form of lexicalized grammar in which the application of syntactic rules is entirely conditioned on the syntactic type, or *category*, of their inputs. No syntactic rule is structure- or derivation- dependent.

Categories identify constituents as either *primitive categories* or *functions*. Primitive categories, such as N, NP, PP, S, and so on, may be regarded as further distinguished by features, such as number, case, inflection, and the like. Functions (such as verbs) bear categories identifying the type of their result (such as S) and that of their argument(s)/complements(s) (both may themselves be either functions or primitive categories). Function categories also define the order(s) in which the arguments must combine, and whether they must occur to the right or the left of the functor. Each syntactic category is associated with a logical form whose semantic type is entirely determined by the syntactic category.

Pure CG (Ajdukiewicz 1935, Bar-Hillel 1953) limits syntactic combination to rules of functional *application* of functions to arguments to the right or left. This restriction limits (weak) expressivity to the level of context-free grammar. However, CCG generalizes the context-free core by introducing further rules for combining categories. Because of their strictly type-driven character and their semantic correspondence to the simplest of the combinators identified by Curry and Feys (1958), these rules are called *combinatory* rules and are the distinctive ingredient of CCG, giving it its name. They are strictly limited to certain directionally specialized instantiations of a very few basic operations, of which the most important are *type-raising* and functional *composition*. A third class of combinatory rules related to *Substitution*, Curry and Feys' **S** combinator, is ignored here.[2]

Though early work in CCG focused primarily on phenomena in English and Dutch, grammar fragments capturing significant cross-linguistic generalizations have been constructed more recently in the framework (e.g., Turkish, Hoffman 1995; Bozsahin 2002; Japanese, Komagata 1999; Tzotzil, Trechsel 2000; Tagalag and Toba Batak, Baldridge 2002).

12.3.1 Categorial Grammar

CCG, like all varieties of Categorial Grammar, eschews context-free production rules like (3). Instead, all language-specific syntactic information is *lexicalized*, via lexical entries like (4):[3]

(3) $S \rightarrow NP\ VP$
$VP \rightarrow TV\ NP$
$TV \rightarrow \{proved,\ finds,\ \ldots\}$

(4) proved $:= (S\backslash NP)/NP$

This syntactic "category" identifies the transitive verb as a function, and specifies the type and directionality of its arguments and the type of its result. We here use the "result leftmost" notation in which a rightward-combining functor over a domain β into a range α are written α/β, while the corresponding leftward-combining functor is written $\alpha\backslash\beta$. α and β may themselves be function categories.[4]

The transitive verb category (4) also reflects its semantic type $(e \rightarrow (e \rightarrow t))$. We can make this semantics explicit by pairing the category with a lambda term, via a colon operator:

(5) proved $:= (S\backslash NP)/NP : \lambda x \lambda y.prove'xy$

(Primes mark constants, non-primes are variables. The notation uses concatenation to mean function application under a “left associative” convention, so that the expression $prove'xy$ is equivalent to $(prove'x)y$.)

In order to capture languages with freer word order, such as Turkish and Tagalog, this notation must be understood as a special case of a more general one allowing categories to be schematized over a number of orders of combination and directionalities. The present paper follows Baldridge (2002) in using a set-CCG notation, according to which the single arguments of a rigid directional category like (4) are replaced by a multiset of one or more argument types, each bearing its own directionality slash, and allowed to combine in any order.

For example, the transitive verb category of a completely free word-order accusative language with nominal case-marking (such as Latin) is written $S|\{NP_{nom}, NP_{acc}\}$, where $|$ indicates that either leftward or rightward combination is allowed for all arguments, and the set brackets indicate that the subject and object can combine in either order. For a language like Tagalog, which is verb-initial but otherwise freely ordered and cased, the corresponding accusative transitive category is written $S/\{NP_{nom}, NP_{acc}\}$. Verb-final Japanese accusative transitives are written $S\backslash\{NP_{nom}, NP_{acc}\}$.

In this extended notation, the English transitive verb can be written in full as $(S\backslash\{NP_{nom}\})/\{NP_{acc}\}$. However, we adopt a convention that suppresses set brackets when argument sets are singletons, so that we continue to write this category as $(S\backslash NP)/NP$, as in (4).

We can generalize the semantic notation introduced at (5) using a parallel argument set notation for lambda terms and a convention that pairs the unordered syntactic arguments with the unordered semantic arguments in the left-to-right order in which they appear on the page. The above transitive verb categories then appear as follows:[5]

(6) a. English: $(S\backslash NP)/NP : \lambda x\lambda y.prove'xy$
b. Latin: $S|\{NP_{nom}, NP_{acc}\} : \lambda\{y,x\}.prove'xy$
c. Tagalog: $S/\{NP_{nom}, NP_{acc}\} : \lambda\{y,x\}.prove'xy$
d. Japanese: $S\backslash\{NP_{nom}, NP_{acc}\} : \lambda\{y,x\}.prove'xy$

All such schemata cover only a finite number of deterministic categories like (4), and can only generate the language that would be generated by compiling out the schema into explicit multiple deterministic lexical categories.[6]

The present paper further follows Jacobson (1990, 1992), Hepple (1990), Baldridge (2002), Baldridge and Kruijff (2003), and Steedman and Baldridge (2006) in assuming that rules and function categories are "modalized," as indicated by a subscript on slashes.[7] Baldridge further assumes that slash modalities are features in a type hierarchy, drawn from some finite set $\mathcal{M}$ (the modalities used here are $\mathcal{M} = \{\star, \diamond, \times, \cdot\}$). The effect of each of these modalities will be described as each of the combinatory rules and its interaction with the modalities is described. The basic intuition is as follows: the $\star$ modality is the most restricted and allows only the most basic applicative rules; $\diamond$ permits order-preserving associativity in derivations; $\times$ allows limited permutation; and $\cdot$ is the most permissive, allowing all rules to apply. The relation of these modalities to each other can be compactly represented via the hierarchy given in figure 12.3.1:[8]

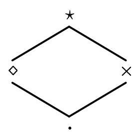


Figure 12.1: CCG type hierarchy for slash modalities (from Baldridge and Kruijff 2003).

We will by convention write the maximally permissive slashes $/_{\cdot}$ and $\backslash_{\cdot}$ as plain slashes $/$ and $\backslash$. This abbreviation again allows us to continue writing the categories that bear this modality, such as the English transitive verb (4), as before.

12.3.2 Combinatory Rules

The simplest operations for combining categories correspond to *functional application*, and can be written as follows:

(7) *The functional application rules*

a. $X/_{\star}Y : f \quad Y : a \quad \Rightarrow \quad X : fa$ (>)

b. $Y : a \quad X\backslash_{\star}Y : f \quad \Rightarrow \quad X : fa$ (<)

Because $\star$ is the supertype of all other modalities, the $/_{\star}$ and $\backslash_{\star}$ slashes on these rules mean that *all* categories can combine by these most basic rules.

The application rules (7) allow derivations equivalent to those of traditional PSCFG, like the following:

(8)
$$\begin{array}{ccc}
\text{Johnson} & \text{met} & \text{Monboddo} \\
\hline
NP : johnson' & (S\backslash NP_{3SG})/NP : \lambda x\lambda y.met'xy & NP : monboddo' \\
 & \multicolumn{2}{c}{\overline{S\backslash NP_{3SG} : \lambda y.met'monboddo'y}^{\,>}} \\
\multicolumn{3}{c}{\overline{S : met'monboddo'johnson'}^{\,<}}
\end{array}$$

Because rules like (7) are written as reductions rather than the traditional productions, such a derivation resembles the selection of a Minimalist Programmatic numeration in the form of an ordered string of lexical items, and the bottom-up construction of a derivation, with those rules performing the function of Merge. It is their operation that projects the head *met′* up the derivation, as in "bare phrase-structural" derivation (2b,c).

Tagalog transitive verbs like *bumili* ("bought") have the following category, in which the λ notation is generalized as in (6) (see Baldridge 2002):

(9) bumili := $S/\{NP_{nom}, NP_{gen}\} : \lambda\{y,x\}.perf'(buy'xy)$

They support multiple word orders, but the derivations are otherwise similar:

(10)
$$\begin{array}{ccc}
\text{Bumili} & \text{ang=babae} & \text{ng=baro} \\
\text{PERF}-\text{AV}-\text{go} & \text{NOM}-\text{woman} & \text{GEN}-\text{dress} \\
\hline
S/\{NP_{nom}, NP_{gen}\} : \lambda\{y,x\}.perf'(buy'xy) & NP_{nom} : woman' & NP_{gen} : dress' \\
\multicolumn{2}{c}{\overline{S/NP_{nom} : \lambda x.buy'x\ woman'}^{\,>}} & \\
\multicolumn{3}{c}{\overline{S : buy'dress'woman'}^{\,>}}
\end{array}$$

(11)
$$\begin{array}{ccc}
\text{Bumili} & \text{ng=baro} & \text{ang=babae} \\
\text{PERF}-\text{AV}-\text{go} & \text{GEN}-\text{dress} & \text{NOM}-\text{woman} \\
\hline
S/\{NP_{nom}, NP_{gen}\} : \lambda\{y,x\}.perf'(buy'xy) & NP_{gen} : dress' & NP_{nom} : woman' \\
\multicolumn{2}{c}{\overline{S/NP_{gen} : \lambda y.buy'dress'y}^{\,>}} & \\
\multicolumn{3}{c}{\overline{S : buy'dress'woman'}^{\,>}}
\end{array}$$

CCG includes a number of further more restricted combinatory operations for merging or combining categories. For present purposes they can be regarded as limited to operations of *type-raising* (corresponding semantically to the combinator **T**) and *composition* (corresponding to the combinator **B**).

Type-raising turns argument categories such as *NP* into functions over the functions that take them as arguments, such as the verbs above, into the results of such functions. Thus NPs like *Johnson* can take on such categories as the following:

(12) a. $S/(S\backslash NP_{3SG}) : \lambda p.p\ johnson'$
b. $S\backslash(S/NP) : \lambda p.p\ johnson'$
c. $(S\backslash NP)\backslash((S\backslash NP)/NP) : \lambda p.p\ johnson'$
d. etc.

This operation must be limited to ensure decidability, and in practice can be strictly limited to argument categories *NP*, *AP*, *PP*, *VP* and *S*. One way to do this is to specify it in the lexicon, in the categories for proper names, determiners, and the like, in which case their original ground types like *NP*, NP/N, etc. can be eliminated.

The inclusion of composition rules like the following as well as simple functional application and lexicalized type-raising engenders a potentially very freely "reordering and rebracketing" calculus, engendering a generalized notion of surface or derivational constituency.

(13) *Forward composition* ($>$**B**)
$X/_{\diamond}Y : f \quad Y/_{\diamond}Z : g \quad \Rightarrow_{\mathbf{B}} \quad X/_{\diamond}Z : \lambda x.f(gx)$

Rule (13) is restricted by the $\diamond$ modality, which means that it cannot apply to categories bearing the $\times$ or $\star$ modalities of Figure 12.3.1.

The inclusion of such rules means that the simple transitive sentence of English has *two* equally valid surface constituent derivations, each yielding the same logical form:

(14)
$$\begin{array}{ccc}
\text{Johnson} & \text{met} & \text{Monboddo} \\
\overline{S/(S\backslash NP_{3SG})}^{>\mathbf{T}} & \overline{(S\backslash NP_{3SG})/NP} & \overline{S\backslash(S/NP)}^{<\mathbf{T}} \\
: \lambda f.f\ johnson' & : \lambda x \lambda y.met'xy & : \lambda p.p\ monboddo' \\
\multicolumn{2}{c}{\overline{S/NP : \lambda x.met'x\ johnson'}^{>\mathbf{B}}} & \\
\multicolumn{3}{c}{\overline{S : met'monboddo'johnson'}^{<}}
\end{array}$$

(15) Johnson met Monboddo

$$\begin{array}{ccc} \text{Johnson} & \text{met} & \text{Monboddo} \\ \overline{S/(S\backslash NP_{3SG})}^{>\mathbf{T}} & \overline{(S\backslash NP_{3SG})/NP} & \overline{(S\backslash NP)\backslash((S\backslash NP)/NP)}^{<\mathbf{T}} \\ :\lambda f.f\ johnson' & :\lambda x\lambda y.met'xy & :\lambda p.p\ monboddo' \end{array}$$

$$\overline{S\backslash NP_{3SG} : \lambda y.met'monboddo'y}^{<}$$

$$\overline{S : met'monboddo'johnson'}^{>}$$

In the first of these, *Johnson* and *met* compose as indicated by the annotation $>\mathbf{B}$ to form a non-standard constituent of type S/NP. In the second, there is a more traditional derivation involving a verbphrase of type $S\backslash NP$. Both yield identical logical forms, and both are legal surface or derivational constituent structures. More complex sentences may have many semantically equivalent derivations, a fact whose implications for processing are discussed in *SP*. (It follows that c-command-dependent phenomena such as binding and control can be (and can only be) captured at the level of logical form (Steedman 1996).)

This theory has been applied to the linguistic analysis of coordination, relativization, and intonational structure in English and many other languages (Steedman 1996, 2000a; Hoffman 1995; Bozsahin 1998; Komagata 1999; Baldridge 1998, 2002). For example, since substrings like *Johnson met* are now fully interpreted derivational *constituents*, complete with compositional semantic interpretations, they can be used to define relativization without movement or empty categories, as in (17), via the following category for the relative pronoun:

(16) that $:= (N\backslash N)/(S/NP) : \lambda p\lambda n\lambda x.(n\ x) \wedge (p\ x)$

(17) (The man) that Johnson met

$$\begin{array}{ccc} \text{that} & \text{Johnson} & \text{met} \\ \overline{(N\backslash N)/(S/NP):} & \overline{S/(S\backslash NP_{3SG}):}^{>\mathbf{T}} & \overline{(S\backslash NP_{3SG})/NP:} \\ \lambda p\lambda n\lambda x.(n\ x) \wedge (p\ x) & \lambda f.f\ johnson' & \lambda x\lambda y.met'xy \end{array}$$

$$\overline{S/NP : \lambda x.met'x\ johnson'}^{>\mathbf{B}}$$

$$\overline{N\backslash N : \lambda n\lambda x.(n\ x) \wedge (met'x\ johnson')}^{>}$$

Such extractions are correctly predicted to be unbounded, since composition can operate across clause boundaries:

(18) (The man) that Johnson says he met

$$\begin{array}{ccccc} \text{that} & \text{Johnson} & \text{says} & \text{he} & \text{met} \\ \overline{(N\backslash N)/(S/NP):} & \overline{S/(S\backslash NP_{3SG}):}^{>\mathbf{T}} & \overline{(S\backslash NP_{3SG})/S:} & \overline{S/(S\backslash NP_{3SG}):}^{>\mathbf{T}} & \overline{(S\backslash NP_{3SG})/NP:} \\ \lambda p\lambda n\lambda x.(n\ x) \wedge (p\ x) & \lambda f.f\ johnson' & \lambda x\lambda y.say'xy & \lambda f.f\ pro' & \lambda x\lambda y.met'xy \end{array}$$

$$\overline{S/S : \lambda x.say'x\ johnson'}^{>\mathbf{B}} \qquad \overline{S/NP : \lambda x.met'x\ pro'}^{>\mathbf{B}}$$

$$\overline{S/NP : \lambda x.say'(met'x\ pro')johnson'}^{>\mathbf{B}}$$

$$\overline{N\backslash N : \lambda n\lambda x.(n\ x) \wedge (say'(met'x\ pro')johnson')}^{>}$$

It is the lexical category (16) of the relative pronoun that establishes the long-range dependency between noun and verb (via the variable x in the present notation). This relation too is established in the lexicon: syntactic derivation merely projects it onto the logical form, with composition and type-raising, as well as application, doing the work of Merge.

Substrings such as *Johnson met* and *Johnson says he met* can also undergo coordination via the following schematized conjunction category, in which '$S\$$' is S or any function category into S, and in the latter case '$\wedge$' schematises over the usual pointwise recursion over logical conjunction (Partee and Rooth 1983):

(19) and := $(S\$\backslash_{\star}S\$)/_{\star}S\$: \lambda p\lambda q.p \wedge q$

This category allows a movement- and deletion- free account of right node raising, as in (20):

(20) [Monboddo likes] and [Johnson says he met] an orangutan

$$\begin{array}{llll}
\text{[Monboddo likes]} & \text{and} & \text{[Johnson says he met]} & \text{an orangutan} \\
\overline{S/NP}^{>\mathbf{B}} & (X\backslash_{\star}X)/_{\star}X & \overline{S/NP}^{>\mathbf{B}} & \overline{S\backslash(S/NP)}^{<\mathbf{T}} \\
 & \multicolumn{2}{l}{\overline{(S/NP)\backslash_{\star}(S/NP)}^{>}} & \\
\multicolumn{3}{l}{\overline{(S/NP)}^{<}} & \\
\multicolumn{4}{l}{\overline{S}^{<}}
\end{array}$$

The $\star$ modality on the conjunction category (19) means that it can *only* combine like types by the application rules (7). Hence, as in GPSG (Gazdar 1981), this type-dependent account of extraction and coordination, as opposed to the standard account using structure-dependent rules, makes the across-the-board condition (ATB) on extractions from coordinate structures (including the "same case" condition) a prediction or theorem, rather than a stipulation, as consideration of the types involved in the following examples will reveal:

(21) a. An orangutan [that$_{(N\backslash N)/(S/NP)}$ [[Johnson met]$_{S/NP}$ and [Monboddo likes]$_{S/NP}$]$_{S/NP}$]$_{N\backslash N}$
b. An orangutan [that$_{(N\backslash N)/(S/NP)}$ *[[Johnson met]$_{S/NP}$ and [likes Monboddo]$_{S/NP}$]$_{S/NP}$]$_{N\backslash N}$
c. An orangutan that$_{(N\backslash N)/(S/NP)}$ *[[Johnson met]$_{S/NP}$ and [Monboddo likes him]$_{S}$]]
d. An orangutan that$_{(N\backslash N)/(S/NP)}$ *[[Johnson met him]$_{S}$ and [Monboddo likes]$_{S/NP}$]

12.3.3 An Aside on some Apparent Exceptions to the Across-the-Board Generalization

Lakoff (1986) has suggested on the basis of examples first noticed by Ross 1967 and Goldsmith 1985 like *What did you go to the store and buy*, *How much beer can you drink and not get sick?*, *This is the stuff that those guys in the Caucasus drink every day and live to be a hundred*, that the coordinate structure constraint and the ATB exception are an illusion. This argument has recently been revived by Kehler (2002) and Asudeh and Crouch (2002). However, it has always also been argued (by Ross and Goldsmith, among others including Lakoff himself in an earlier incarnation) that these extractions involve another, non-coordinate, subordinating lexical category for “and”, and as such do not constitute counterexamples to the CSC and ATB constraints after all. Among the arguments in support of this view are the presuppositional and volitional semantics of the sentences in question (and the absence of such overtones from true coordinates), and the fact that (as Postal 1998 points out), no *other* conjunctions support such extractions—cf. **What did you go to the store or buy*, **How much beer can you drink or not get sick?*, **This is the stuff that those guys in the Caucasus drink every day or live to be a hundred*. Moreover, the ATB-violating leftward extractions are not in general mirrored by equivalent right node raising, unlike the across-the-board cases such as (20):

(22) *Those guys in the Caucasus drink every day and live to be a hundred a kind of fermented mare’s milk.

It follows that the problematic extractions can naturally be handled in CCG by assigning to “and” additional independent categories supporting extraction from left and right conjunct respectively, and with an appropriate volitional/causal semantics (omitted here) of the kind discussed by Lakoff, Kehler, and Asudeh and Crouch, here written as follows:[9]

(23) a. and := $((VP/NP_{+ANT})\backslash_{\star}(VP/NP))/_{\star}VP$
b. and := $((VP/NP)/_{\star}(VP/NP))\backslash_{\star}VP$

The possibility of such exceptional extractions therefore does not controvert the CCG or GPSG claims that the coordinate structure structure constraint and its exceptions arise from the “same types” requirement on coordinands, contrary to claims by these authors.

12.4 Quantifier Scope

One might expect that the generalized notion of surface derivation afforded by CCG, as illustrated in (14) and (15), in which the object and the subject respectively command the rest of the sentence, could be exploited to explain the fact that multiply quantified sentences like the following appear to have multiple logical forms in which either quantifier may outscope the latter, without any appeal to quantifier raising or covert movement.

(24) Everyone loves someone. $(\forall\exists/\exists\forall)$

(25) Someone loves everyone. $(\forall\exists/\exists\forall)$

However, this cannot be correct. Sentences like the following have only *one* CCG analysis, in which the right node raised object commands everything including the subjects:

(26) Every boy likes and every girl hates some novelist.

Nevertheless, such sentences have a reading in which novelists are dependent on boys and girls, as well as a reading where there is just one novelist.

However, any temptation to allow covert quantifier movement at the level of logical form should be resisted, for two reasons. First, under present Montagovian assumptions concerning transparency of syntax to semantics, having shown the appearance of overt syntactic movement to be an illusion, it would be perverse to then postulate a kind of movement to which syntax is *not* transparent. Second, the available readings for (26) are subject to a parallelism restriction first noted by Geach (1972). That is to say that, while there is a reading under which the novelists are all dependendent, and a reading in which there is a single novelist that everyone either likes or hates, there are no mixed readings such as the one where the boys all like the same novelist but the girls all hate different ones. There is not even a reading where there is one novelist that the boys all like and a different novelist that the girls all hate.

This restriction *does* appear to reflect the CCG derivation, in the sense that *some novelist* has to either take wide or narrow scope with respect to the *entire* residue of right node raising, a fact that is hard to explain if quantifiers are free to covertly move independendently.

In *SP*, Geach's observation is explained on the assumption that the so-called existential quantifier determiners are not quantifier determiners at all, but rather are determiners of Skolem terms, which consist of a Skolem functor applied to all variables bound by universal quantifiers in whose scope the Skolem term falls. Thus, the categories for universals and existentials look quite different semantically. The universals are traditional generalized quantifiers, e.g.:

(27) every := $(S/(S\backslash NP))/N : \lambda n\lambda p.\forall x[nx \wedge px]$
every := $((S\backslash NP)\backslash((S\backslash NP)/NP)/N : \lambda n\lambda p\lambda y.\forall x[nx \wedge pxy]$
...

The logical form in such categories does the work of "covert quantifier raising", by giving the universal quantifier scope over the clause and relating it to an lf-commanded position via the bound variable. However this is done lexically and statically, without movement or structural change.

By contrast, the existentials are not generalized quantifiers, but "unspecified" Skolem individuals, e.g.:

(28) some := $(S/(S\backslash NP))/N : \lambda n\lambda p.p(skolem'n)$
some := $((S\backslash NP)\backslash((S\backslash NP)/NP)/N : \lambda n\lambda p.p(skolem'n)$
...

This ensures that for both the left-branching derivation exemplified in (14) and the right-branching ones like (15), we get both "wide" and "narrow scope" readings for existentials in sentences like (24). For example, the following are the two readings for the former, left-branching, derivation (those for the latter, more standard, right-branching derivation are suggested as an exercise).

(29)

Everyone	loves	someone
$S/(S\backslash NP_{3SG})$	$(S\backslash NP_{3SG})/NP$	$S\backslash(S\backslash NP)$
$: \lambda p.\forall y[person'y \rightarrow py]^{\{y\}}$	$\lambda x.\lambda y.love'xy$	$: \lambda q.q(skolem'person')$

$$\frac{}{S/NP : \lambda x.\forall y[person'y \rightarrow love'xy]^{\{y\}}} >\mathbf{B}$$

$$\frac{}{S : \forall y[person'y \rightarrow love'(skolem'person')y]^{\{y\}}} <$$

$$\cdots\cdots\cdots$$

$$S : \forall y[person'y \rightarrow love'sk^{(y)}_{person'}y]^{\{y\}}$$

(30)
$$
\begin{array}{ccc}
\text{Everyone} & \text{loves} & \text{someone} \\
\hline
S/(S\backslash NP_{3SG}) & (S\backslash NP_{3SG})/NP & S\backslash(S\backslash NP) \\
: \lambda p.\forall y[person'y \rightarrow py]^{\{y\}} & \lambda x.\lambda y.love'xy & : \lambda q.q(skolem'person') \\
\hline
\multicolumn{2}{c}{S/NP : \lambda x.\forall y[person'y \rightarrow love'xy]^{\{y\}} \quad {>}\mathbf{B}} & \cdots\cdots\cdots \\
 & & : \lambda q.q(sk_{person'}) \\
\hline
\multicolumn{3}{c}{S : \forall y[person'y \rightarrow love'(sk_{person'})y]^{\{y\}} \quad {<}}
\end{array}
$$

In (30), the skolem term indefinite is a constant, rather than a function term in the bound variable *y* in its environment.

Because universals, by contrast with existentials, are genuine quantifiers, they and they alone can truly invert scope in both right- and left-branching derivations. For example, *every* can invert as follows in sentence (25)(once again the left-branching inverting reading and the non-inverting readings for both derivations are suggested as an exercise):

(31)
$$
\begin{array}{ccc}
\text{Someone} & \text{loves} & \text{everyone} \\
\hline
S/(S\backslash NP_{3SG}) & (S\backslash NP_{3SG})/NP & (S\backslash NP)\backslash((S\backslash NP)/NP) \\
: \lambda p.p(skolem'person') & : \lambda x\lambda y.love'xy & : \lambda q.\forall x[person'x \rightarrow qx]^{\{x\}} \\
\hline
 & \multicolumn{2}{c}{S\backslash NP_{3SG} : \lambda y.\forall x[person'x \rightarrow love'xy]^{\{x\}} \quad {<}} \\
\hline
\multicolumn{3}{c}{S : \forall x[person'x \rightarrow love'x(skolem'person')]^{\{x\}} \quad {>}} \\
\cdots\cdots\cdots & \cdots\cdots\cdots & \cdots\cdots\cdots \\
\multicolumn{3}{c}{S : \forall x[person'x \rightarrow love'x\ sk^{(x)}_{person'}]^{\{x\}}}
\end{array}
$$

Similar derivations allow the universals *every*, *each*, and free-choice *any* to invert over most non-universals, such as *(at least/exactly/at most) two* and *several, many, most*.

Crucially, the formation of such Skolem terms can occur before reduction of the object and the prefix in (26), in which case there are no scoping universal quantifiers and the Skolem term is a constant appearing to "take scope everywhere". But the Skolem term can also be formed after the reduction, in which case each copy of the Skolem term at the level of logical form is obligatorily dependent on its lf-commanding universal. Hence, the missing readings are excluded without appeal to otherwise unmotivated "parallelism constraints" on coordinate structures (Goodall 1987).

A number of other curious freedoms and restrictions, such as anomalous scope properties of "donkey sentences" and the non-ability of non-universals in sentences like (32) to invert scope in the strong sense of distributing over c-commanding existentials as universals do (cf. (25)), are explained by the treatment of existentials as Skolem terms rather than generalized quantifiers.

(32) Some linguist knows at least three languages. (∃3/#3∃)

This account is further developed in Steedman (2005b).

12.5 Universal Grammar

Even quite small sets of functional combinators, including the set {**BTS**} implicit in CCG, can yield calculi of the full expressive power of Turing machines and the simply typed λ calculus. However, CCG syntax is subject to a number of principles which make it weakly equivalent to TAG and Linear Indexed Grammar (LIG, Aho 1968; Gazdar 1988), the least more powerful natural class of languages than CFG that is known, chacterized by a generalization of the push-down automaton (PDA), the Embedded PDA (EPDA) (Vijay-Shanker and Weir 1990, Joshi, Vijay-Shanker and Weir 1991, Vijay-Shanker and Weir 1993, Vijay-Shanker and Weir 1994). This means that the theory of the computation in Marr's sense is very low power, only just trans- context-free. This equivalence gives rise to a polynomial time worst-case complexity result, and means that standard CF parsing algorithms such as CKY (Cocke and Schwartz 1970) and standard probabilitically optimizing parsing models such as head-dependency models (Collins 1997) immediately generalize to CCG (Steedman 2000b). Such grammars and models have been successfully applied to wide-coverage parsing of the Penn Wall Street Journal corpus by Hockenmaier and Steedman (2002), Clark, Hockenmaier and Steedman (2002), Hockenmaier (2003), Clark and Curran (2004), and Clark, Steedman and Curran (2004), with state-of-the-art levels of recovery of semantically significant dependencies.

The principles which limit the power of combinatory rules in this way can be summed up as a "projection principle" which says that syntax must project, and may not override, the directional information specified in the lexicon, and, conversely, that the lexicon should not do syntax's job of unbounded projection. This principle is expressed in the next section as a number of subsidiary principles.

12.5.1 The Combinatory Projection Principle

We have given examples of several rules that encode the syntactic reflex of a few basic semantic functions (combinators). However, a larger set of possible rules could be derived from the combinators, and we restrict the set to be only those which obey the following principles:

(33) *The Principle of Adjacency:*
Combinatory rules may only apply to finitely many phonologically realized and string-adjacent entities.

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

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

The first of these principles is merely the definition of combinators themselves. The other principles say that combinatory rules may not override, but must rather "project," the directionality specified in the lexicon. More concretely, the Principle of Consistency excludes the following kind of rule:

(36) $X\backslash_{\star}Y \quad Y \quad \Rightarrow \quad X$ (disallowed)

The Principle of Inheritance excludes rules like the following hypothetical instances of composition:

(37) a. $X/_{\diamond}Y \quad Y/_{\diamond}Z \quad \Rightarrow \quad X\backslash_{\diamond}Z$ (disallowed)
b. $X/_{\diamond}Y \quad Y/_{\diamond}Z \quad \Rightarrow \quad X/_{\times}Z$ (disallowed)

On the other hand, these principles do allow rules such as the following:

(38) *The crossing functional composition rules*
a. $X/_{\times}Y \quad Y\backslash_{\times}Z \quad \Rightarrow \quad X\backslash_{\times}Z$ $(>\mathbf{B}_{\times})$
b. $Y/_{\times}Z \quad X\backslash_{\times}Y \quad \Rightarrow \quad X/_{\times}Z$ $(<\mathbf{B}_{\times})$

Such rules are not theorems of type calculi such as that of Lambek (1958) and its descendants, and in fact cause collapse of such calculi into permutation completeness if added as axioms (Moortgat 1988), a fact that has motivated the development of multi-modal varieties of categorial grammar within the type-logical tradition by Hepple (1990), Morrill (1994), and Oehrle (2000).

While such rules do not cause a collapse in CCG *even without the modalities*, the present use of modalities to provide finer control over the rules is directly inspired by multi-modal categorial grammar (see Baldridge 2002). They must be restricted by the $\times$ modality, which is incompatible with $\star$ and $\diamond$ modalities, because they have a re-ordering effect.

The composition rules are all generalized to cover cases where the "lower" function $Y|Z$ is of higher valency $(Y|Z)|W$, etc., up to some low value such as 4 $(((Y|Z)|W)|V)|U$, which appears to be the highest valency in the lexicon. It is the combination of crossed composition and generalized composition rules that increases the expressive power of the formalism to the lowest trans-context-free level of the "mildly context-sensitive" class identified by Joshi, Vijay-Shanker and Weir (1991).

12.5.2 The Categorial Lexicon

The lexicon of a given language is a finite subset of the set of all categories subject to quite narrow restriction that ultimately stem from limitations on the variety of semantic types with which the syntactic categories are paired in the lexicon. In particular, we can assume that lexical function categories are limited to finite—in fact, very small—numbers of arguments. (For English at least, the maximum appears to be four, required for a small number of verbs like *bet*, as in *I bet you five dollars I can spit further than you.*)

The most basic assumption of the present approach is that the responsibility for specifying all dependencies, whether long-range or local, resides in the lexical specifications of syntactic categories for the "heads" of those dependencies—that is, the words corresponding to predicate-argument structural functors, such as verbs. This principle, which is related to the Projection Principle of GB, can be more formally stated as follows:[10]

(39) *The Principle of Lexical Head Government*
Both bounded and unbounded syntactic dependencies are specified by the lexical syntactic type of their head.

This is simply to say that the present theory of grammar is "lexicalized," a property that makes it akin to LFG, TAG, Head-Driven Phrase Structure Grammar (HPSG; Pollard and Sag 1994), and certain recent versions of GB (see Hale and Keyser 1993; Brody 1995, and Chomsky and Lasnik in Chomsky 1995b:25).

Lexicalized grammars make the lexical entries for words do most of the grammatical work of mapping the strings of the language to their interpretations. The size of the lexicon involved is therefore an important measure of a grammar's complexity. Other things being equal, one lexical grammar is simpler than another if it captures the same pairing of strings and interpretations using a smaller lexicon.

A more distinctive property of CCG, which it shares with LFG and GB, and which sets it apart from TAG, GPSG, and HPSG (which in other respects are more closely related), is that it attempts to minimize the size of the lexicon by adhering as closely as possible to the following stronger principle:

(40) *The Principle of Head Categorial Uniqueness*
A single nondisjunctive lexical category for the head of a given construction specifies both the bounded dependencies that arise when its complements are in canonical position and the unbounded dependencies that arise when those complements are displaced under relativization, coordination, and the like.

That is not to say that a given word may not be the head of more than one construction and hence be associated with more than one category. Nor (as we have seen for the case of Tagalog) does it exclude the possibility that a given word-sense pair may permit more than one canonical order, and hence have more than one category per sense, possibly schematized using the set-CCG notation of (6). The claim is simply that each of these categories specifies both canonical order and all varieties of extraction for the clause type in question. For example, a single lexical syntactic category (5) for the word *met*, which does not distinguish between "antecedent," "θ," or any other variety of government, is involved in all of the dependencies illustrated in (8), (17), (18), and (20).

By contrast, in both TAG and GPSG these dependencies are mediated by different initial trees or categories, and in HPSG they are mediated by a disjunctive category.

Unlike the principles defined earlier, exceptions to the Principle of Head Categorial Uniqueness are sometimes forced.[11] However, each such exception complicates the grammar by expanding the lexicon, and makes it compare less favorably with an otherwise equivalently valued grammar that requires no such exceptions. Hence, such exceptions are predicted to be rare.

Many remaining open questions in both CCG and MP concern the notion "possible lexical category". Many of them are addressed by MP principles like Full Interpretation and Shortest Move. Since the move from $\overline{X}$ theory to "Bare Phrase Structure" theory (Chomsky 1995b,a) exemplified in (2) looks very much like a move to a categorial, rather than phrase-structure, base grammar, it is natural to look for convergence.

Such principles look rather different when viewed from the CCG perspective. For example, it is Shortest Move that in MP terms allows bounded raising in (a) below, and also (via A-chain formation on the controlled subjects) in (b), whilst disallowing unbounded raising in (c) to deliver a reading where it is likely that Monboddo seems to be happy:[12]

(41) a. Monboddo seems to be happy.
     b. Monboddo is likely to seem to be happy
     c. *Monboddo is likely that it seems to be happy.

The raising predicates *seems* and *likely* have the following categories (as well as categories (43), supporting the expletive predications):

(42) a. seems/seem := $(S\backslash NP)/(S_{to}\backslash NP) : \lambda p\lambda x.seemingly'(px)$
     b. likely := $(S_{pred}\backslash NP)/(S_{to}\backslash NP) : \lambda p\lambda x.probably'(px)$

(43) a. seems := $(S\backslash NP_{it})/S_{CP} : \lambda s\lambda x.seemingly's$
     b. likely := $(S_{pred}\backslash NP_{it})/S_{CP} : \lambda s\lambda x.probably's$

Thus *likely to seem to be happy* in (41b) is derived as follows (combination of the copula and *to* is suppressed to aid readability):

(44)

$$\dfrac{\dfrac{\dfrac{\text{likely}}{(S_{pred}\backslash NP)/(S_{to}\backslash NP) : \lambda p\lambda x.probably'(px)} \quad \dfrac{\text{to seem}}{(S_{to}\backslash NP)/(S_{to}\backslash NP) : \lambda p\lambda x.seemingly'(px)}}{(S_{pred}\backslash NP)/(S_{to}\backslash NP) : \lambda p\lambda x.probably'(seemingly'(px))}{>}\mathbf{B} \quad \dfrac{\text{to be happy}}{S_{to}\backslash NP : happy'}}{S_{pred}\backslash NP : \lambda x.probably'(seemingly'(happy'x))}{>}$$

Of course, these categories do not allow any analysis for (41c). Moreover, the only possibility for obtaining the intended reading is to give *likely* a completely unrelated relative pronoun-like category analogous to a tough-movement predicate like *easy* (see Steedman 1996:29,62), and to give *seems* a category analogous to that of a subject-extracting bare complement verb like *thinks* (ibid.:58). One way one might think of doing this is as follows:

(45) *likely := $(S_{pred}\backslash NP)/(S_{CP}/NP) : \lambda p\lambda x.probably'(px)$

(46) *seems := $((S\backslash NP_{it})/NP_{+ANT})/(S_{to}\backslash NP) : \lambda p\lambda x\lambda y.seemingly'p(x)$

Derivation of (41c) could then proceed analogously to subject extraction (Steedman 1996:58).[13]

However, these categories will immediately overgeneralize to other unbounded dependencies, allowing relativization, tough movement, and other absurdities:

(47) a. *A woman who it seems to be happy.
b. *Monboddo is easy to believe that it seems to be happy.
c. *Monboddo is likely that Johnson likes
d. *Monboddo is likely that Johnson thinks likes him

It might seem that, as a last resort, we could rewrite the above categories as follows, using categories with a feature unique to the Shortest Move-violating categories—call it *FIXIT*—and introducing a parallel restriction $-FIXIT$ on the relative pronoun and tough-movement categories to prevent examples like (47):

(48) *likely := $(S_{pred}\backslash NP)/(S_{CP}/NP_{+FIXIT}) : \lambda p\lambda x.probably'(px)$

(49) *seems := $((S\backslash NP_{it})/NP_{+FIXIT})/(S_{to}\backslash NP) : \lambda p\lambda x\lambda y.seemingly'p(x)$

However, such a move amounts to introducing an entirely new species of lexically specified unbounded dependency into the grammar just for this construction. Moreover, both (48) and the revised relative pronoun and tough-movement categories would be in violation of the Principle of Lexical Head Government (39), for these categories are not the head of the dependency that they mediate. Such categories have no parallel elsewhere in the grammar of English or any other language.

Thus, Shortest Move as it applies to raising and control appears to be a consequence of categorially lexicalizing the grammar, rather than an active principle of the theory of grammar in its own right.

12.6 Three Case Studies in CCG

The following sections briefly exemplify some of the constructions that have presented difficulties for other theories of grammar, and which are successfully accomodated by CCG. The reader is referred to the literature already cited for fuller accounts of these and other analyses.

12.6.1 Case Study 1: Argument Cluster Coordination

CCG has been used to analyze a wide variety of coordination phenomena, including English “argument-cluster coordination”, “backward gapping” and “verb-raising” constructions in Germanic languages, and English gapping. The first of these is illustrated by the following analysis, from Dowty (1988—cf. Steedman 1985), in which the ditransitive verb category$(VP/NP)/_{\diamond}NP$ is abbreviated as *DTV*, and the transitive verb category VP/NP is abbreviated as *TV*:[14]

(50) give Walt the salt and Malcolm the talcum

- give: DTV
- Walt: $TV\backslash_{\diamond}DTV$ ($<\mathbf{T}$)
- the salt: $VP\backslash TV$ ($<\mathbf{T}$)
- and: $(X\backslash_{\star}X)/_{\star}X$
- Malcolm: $TV\backslash_{\diamond}DTV$ ($<\mathbf{T}$)
- the talcum: $VP\backslash TV$ ($<\mathbf{T}$)
- Walt the salt: $VP\backslash_{\diamond}DTV$ ($<\mathbf{B}$)
- Malcolm the talcum: $VP\backslash_{\diamond}DTV$ ($<\mathbf{B}$)
- and Malcolm the talcum: $(VP\backslash_{\diamond}DTV)\backslash_{\star}(VP\backslash_{\diamond}DTV)$ ($>$)
- Walt the salt and Malcolm the talcum: $VP\backslash_{\diamond}DTV$ ($<$)
- give Walt the salt and Malcolm the talcum: VP ($<$)

Since we have assumed the previously discussed rules of forward type-raising ($>\mathbf{T}$) and forward composition ($>\mathbf{B}$), this construction is correctly predicted to exist in English by arguments of symmetry, which imply that their backward varieties, $<\mathbf{T}$ and $<\mathbf{B}$ must also be assumed.

Given independently motivated limitations on type-raising, examples like the following are still disallowed:[15]

(51) *Three mathematicians [[in ten]$_{PP}$ [derive a lemma,]$_{S\backslash NP}$] and [in a hundred prove completeness.]

12.6.2 Case Study 2: English Intonation and Information Structure

We also have seen that, in order to capture coordination with rules adhering to the constituent condition, CCG generalizes surface constituency to give substrings like *Marcel proved* and even *Malcolm the talcum* the full status of constituents.

But if they are constitutents of coordinate constructions, they are predicted to be possible constituents of ordinary non-coordinate sentences as well. The characteristics of English intonation structure show that this prediction is correct.

Consider the following minimal pair of dialogs, in which intonational tunes are indicated both informally via parentheses and small capitals as before, and in the standard notation of Pierrehumbert (1980) and Pierrehumbert and Beckman (1988), in which prosodic phrases are specified solely in terms of two kinds of elements, the pitch accent(s) and the boundary:

(52) Q: I know who proved soundness. But who proved COMPLETENESS?
A: (MARCEL) (proved COMPLETENESS).
H* L L+H* LH%

(53) Q: I know which result Marcel PREDICTED. But which result did Marcel PROVE?
A: (Marcel PROVED) (COMPLETENESS).
L+H*LH% H* LL%

In (52A), there is a prosodic phrase on MARCEL including the sharply rising pitch accent that Pierrehumbert calls H*, immediately followed by an L boundary, perceived as a rapid fall to low pitch. There is another prosodic phrase having the somewhat later-rising and (more importantly) lower-rising pitch accent called L+H* on *COMPLETENESS*, preceded by null tone (and therefore interpolated low pitch) on the word *proved* and immediately followed by an utterance-final rising boundary, written LH%.

In (53A) above, the order of the two tunes is reversed: this time, the tune with pitch accent L+H* and boundary LH% occurs on the word *PROVED* in one prosodic phrase, *Marcel PROVED*, and the other tune with pitch accent H* and boundary LL% is carried by a second prosodic phrase *COMPLETENESS*.[16]

The intuition that these tunes strongly convey systematic distinctions in discourse meaning is inescapable. (For example, exchanging the answer tunes between the two contexts in (52) and (53) makes the answers completely incomprehensible.) Prevost and Steedman (1994) claim that the tunes L+H* LH% and H* L (or H* LL%) are respectively associated with the "theme" and “rheme” of the sentence, where these terms are used in the sense of Mathesius (1929), Firbas (1964, 1966), and Bolinger (1989), and correspond roughly to a generalization of the more familiar terms “topic” and “comment”, which however are generally restricted by definition to traditional constituents.

Informally the theme can be thought of as corresponding to the content of a contextually available *wh*-question, which may be explicit, as in (52) and (53), or implicit in other discourse content. The position on the pitch accent, if any, in the theme, distinguishes words corresponding to “focused” elements of the content which distinguish this theme from other contextually available alternatives. The rheme can then be thought of as providing the answer to the implicit *wh*-question, with the pitch accent again marking focused words which distinguish this answer semantically from other potential answers. The system comprising the oppositions of theme/rheme and focus/background is known as information structure. Steedman 2000a provides a more formal definition in terms of the “alternative semantics” of Rooth (1985, 1992), and the related “structured meanings” of Cresswell (1973, 1985), von Stechow (1991), and others, which Steedman 2006 grounds in notions of common ground and common ground update.[17]

The fact that CCG allows alternative derivations like (14) and (15) offers an obvious way to bring intonation structure and its interpretation – information structure – into the same syntactic system as everything else: Crucially, these alternative derivations are guaranteed to yield the same predicate argument relations, as exemplified by the logical form that results from the two derivations. However, the derivations build this logical form via different routes that construct lambda terms corresponding semantically to the theme and rheme. In particular the derivation (15) corresponds to the information structure associated with the intonation contour in (52), while derivation (14) corresponds to that in (53).

This observation can be captured by making pitch accents mark both arguments and results of CCG lexical categories with theme/rheme markers θ/ρ, as in the following category for a verb bearing an L+H* accent:

(54) proved := $(S_\theta \backslash NP_\theta)/NP_\theta : \lambda x \lambda y. *prove'xy$

The predicate is marked as focused or contrasted by the * marker in the logical form. θ/ρ marking is projected onto the arguments and result of constituents by combinatory derivation. The boundary tones like LH% have the effect of completing information structural constituents, and transfering theme/rheme marking to θ'/ρ' marking to constituent interpretations at logical form, as indicated in outline in Figure 12.8. We will pass over further details of exactly how this works, referring the reader to Prevost (1995) and to Steedman (2000a, 2006). The latter papers generalize this approach to the full range of tunes identified by Pierrehumbert, including those with multiple pitch accents and multiple or disjoint themes and rhemes.

12.6.3 Case Study 3: Crossing Dependencies

The availability of *crossed* composition (38) under the Principles of Consistency and Inheritance allows crossing dependencies in Dutch and certain Swiss dialects of German, which cannot be captured by CFG and have given rise to proposals for "verb-raising" transformational operations, as in the following example (from Shieber):

(55)

das	mer	em Hans	es huus	hälfed	aastriiche
that	we−NOM	Hans−DAT	the house−ACC	helped	paint
	$NP^{\uparrow}_{nom}$	$NP^{\uparrow}_{dat}$	$NP^{\uparrow}_{acc}$	$((S_{+SUB}\backslash NP_{nom})\backslash NP_{dat})/VP$	$VP\backslash NP_{acc}$

$$\frac{((S_{+SUB}\backslash NP_{nom})\backslash NP_{dat})/VP \quad VP\backslash NP_{acc}}{((S_{+SUB}\backslash NP_{nom})\backslash NP_{dat})\backslash NP_{acc}} >\mathbf{B}_\times$$

$$\frac{NP^{\uparrow}_{acc} \quad ((S_{+SUB}\backslash NP_{nom})\backslash NP_{dat})\backslash NP_{acc}}{(S_{+SUB}\backslash NP_{nom})\backslash NP_{dat}} >$$

$$\frac{NP^{\uparrow}_{dat} \quad (S_{+SUB}\backslash NP_{nom})\backslash NP_{dat}}{S_{+SUB}\backslash NP_{nom}} >$$

$$\frac{NP^{\uparrow}_{nom} \quad S_{+SUB}\backslash NP_{nom}}{S_{+SUB}} >$$

"that we helped Hans paint the house"

The universal $\cdot$ modality on the verbs *hälfed* and *aastriichte* (suppressed as usual by convention) permits the forward crossed composition rule (38a) to apply. (The corresponding categories for the more rigid word-order Dutch are restricted by $\times$ modality, see Baldridge 2002 and section 12.7.1, below) The tensed verb is distinguished as the head of a subordinate clause via the feature SUB. The type-raised NP categories are abbreviated as $NP^{\uparrow}_{case}$, since the fact that they are raised is not essential to understanding the point about crossing dependencies. It is correctly predicted that the following word orders are also allowed in at least some dialects (Shieber 1985:338-9):

(56) a. a. das mer em Hans hälfed es huus aastriiche.
b. b. das em Hans mer es huus hälfed aastriiche.

The construction is completely productive, so the dependencies are not only intersective, but unbounded. For example, we have the following (also from Shieber):

(57)

das	mer	d' chind	em Hans	es huus	lönd	hälfe	aastriiche
that	we−NOM	the children−ACC	Hans−DAT	the house−ACC	let	help	paint
	$NP^{\uparrow}_{nom}$	$NP^{\uparrow}_{acc}$	$NP^{\uparrow}_{dat}$	$NP^{\uparrow}_{acc}$	$((S_{+SUB}\backslash NP_{nom})\backslash NP_{acc})/VP$	$(VP\backslash NP_{dat})/VP$	$VP\backslash NP_{acc}$

$$\text{lönd hälfe} \Rightarrow_{>B^2_\times} (((S_{+SUB}\backslash NP_{nom})\backslash NP_{acc})\backslash NP_{dat})/VP$$
$$\text{lönd hälfe aastriiche} \Rightarrow_{>B_\times} (((S_{+SUB}\backslash NP_{nom})\backslash NP_{acc})\backslash NP_{dat})\backslash NP_{acc}$$
$$\text{es huus} \ldots \Rightarrow_{>} ((S_{+SUB}\backslash NP_{nom})\backslash NP_{acc})\backslash NP_{dat}$$
$$\text{em Hans} \ldots \Rightarrow_{>} (S_{+SUB}\backslash NP_{nom})\backslash NP_{acc}$$
$$\text{d' chind} \ldots \Rightarrow_{>} S_{+SUB}\backslash NP_{nom}$$
$$\text{mer} \ldots \Rightarrow_{>} S_{+SUB}$$

"that we let the children help Hans paint the house"

Again, the unbounded dependencies are projected from the lexical frame of the verb, without syntactic movement, and by the same category that supports the non-verb-raised cases (56).

12.7 CCG as a Minimalist Generative Grammar: $a^n b^n$ Crossing

At first glance, because it has been presented as a system of reductions rather than productions, this theory might not look like a truly *generative* grammar. In fact, like standard generative grammars, it is neutral with respect to the direction of application of the rules, and can be translated into an equivalent set of productions in a number of ways. The easiest (but the least helpful in computational terms) is the one that seems to be assumed by Chomsky for the Miminalist Program—that is, to enumerate all possible strings or numerations, and use CCG to decide which numerations are sentences of the language (cf. Chomsky 1995b:227,237).

However, a more attractive alternative is to write a traditional base-generative grammar of productions for CCG logical forms, along the lines of a base generative grammar for S-structures. To do this over the *normalized* logical forms is in fact the first step in providing a model theory for LF, as proposed in Steedman 2005b as part of the CCG-based account of natural language quantifier scope alternation described in section 12.4. However, the problem of mapping such logical forms onto phonological forms is not simple, and reintroduces many of the problems of movement and copy-detection that CCG was designed to eliminate. What we need to do instead is to turn the grammar into into a system of productions for generating both *unnormalized* logical forms, which we have seen corresond at the top level to information structures, and syntactic types, in parallel. We can do this directly by building an equivalent Linear Indexed Grammar, compiling out the combinatory reductions into an equivalent set of productions. The next sections illustrates this process for a small artificial grammar for the language $a^n b^n$ with crossing dependencies, analogous to (but not isomorphic to) the CCG grammar for the Swiss German construction discussed in section 12.6.3. To make the grammar more linguistically intelligible, Dutch proper names are used for the *a*s and Dutch verbs are used for the *b*s (cf. Steedman 1985)

In order to simplify the translation to LIG, this version replaces λ-terms with a related notation derived from prolog unification, in which logical forms in productions are specified in terms of variables, which become instantiated when the leaves are unified with lexical items.

Capitals are variables over syntactic or semantic terms, and lower case symbols are constants. $X\hat{}P$ simulates $\lambda x.p$ where p is a variable over terms in x. The grammar can be directly realized as a Prolog Definite Clause Grammar (DCG), although some care is needed because of the left recursive rules.

12.7.1 The Lexicon:

This is identical to the CCG lexicon (although it is simpler than the one in section 12.6.3).

(58) $\mathit{sinf}\backslash np : X\hat{}\mathit{fall}'(X) \rightarrow$ fallen (“fall”)
$(s\backslash np)/_{\times}\mathit{sinf} : P\hat{}X\hat{}\mathit{see}'(X,P) \rightarrow$ zag (“saw”)
$(\mathit{sinf}\backslash np/_{\times})\mathit{sinf} : P\hat{}X\hat{}\mathit{see}'(X,P) \rightarrow$ zien (“see”)
$np : \mathit{harry}' \rightarrow$ Hendrik
$np : \mathit{cecilia}' \rightarrow$ Cecilia
$np : \mathit{jan}' \rightarrow$ Jan
$\ldots \mathit{etc.}$

$\times$ modality ensures that only rules corresponding to *crossed* functional composition will be generated. composition will be generated

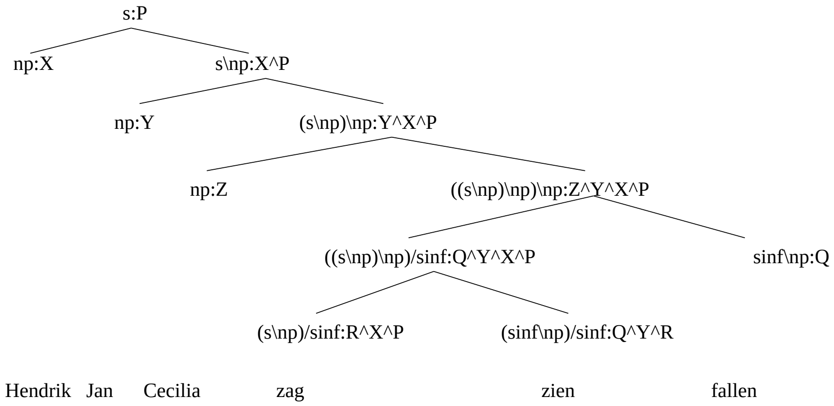


Figure 12.2
Before Lexical Insertion

12.7.2 The Rules

The following linear indexed rules are generated using the same parameters as the lexicon from the universal set of combinatory rules. The variable S, matches arbitrarily high valency syntactic categories and acts as the linear indexed grammar stack-valued feature (note that it passes to exactly one daughter). The first rule recursively generates n NPs and an n-ary verb category with function application as its semantics. The second rule generates an $n-1$-ary verb and the nth *fallen*-type verb. The third rule recursively generates a sequence of one *zag*- and $n-2$ *zien* type verbs. group, with function composition semantics.

(59) $S : P \rightarrow np : X \quad S\backslash np : X\hat{}P$
$(S\backslash np)\backslash np : Y\hat{}X\hat{}P \rightarrow S\backslash np/sinf : Q\hat{}X\hat{}P \quad sinf\backslash np : Y\hat{}Q$
$(S\backslash np)/sinf : Q\hat{}X\hat{}P \rightarrow S/sinf : Q\hat{}P \quad (sinf\backslash np)/sinf : R\hat{}X\hat{}Q$

This grammar thus generates all and only the strings of the “verb raising” trans-context-free Dutch fragment NP^n *zag* $zien^{n-2}$ *fallen* with n crossed dependencies for $n \geq 2$.

12.7.3 “The Computation” and “Spell-out”

The tree in Figure 12.2 shows the generation of *(dat) Hendrik Jan Cecilia zag zien fallen* (“that Harry saw Jan see Cecilia fall”) with crossed dependencies,

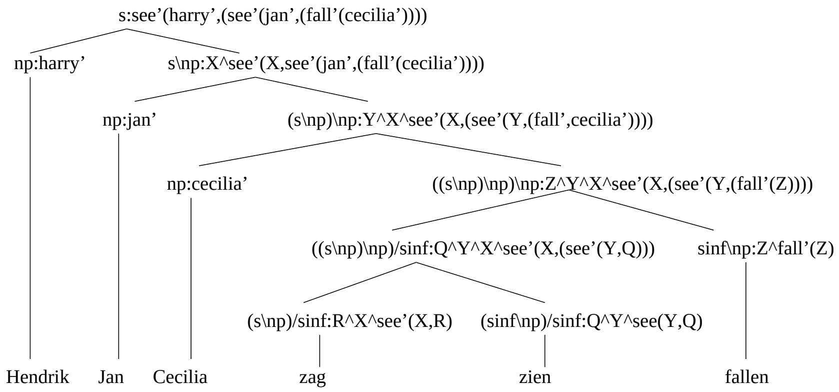


Figure 12.3
After Lexical Insertion

up to the stage where all leaves are preterminals, with corresponding lexical entries, before lexical insertion or spell-out:

Such trees are fully ordered. They define a set of ordered numerations or strings, all of which are legal strings of the language, and all of which support the analysis implicit in the tree, and its (unordered, and as yet massively underspecified) LF.

The effect of instantiating the preterminals in such trees with lexical items is to instantaneously project specific unordered logical forms onto the entire hitherto underspecified derived logical form. This process is illustrated in a simplified form in Figure 12.3:[18]

Specifying the tree in this way selects one string or numerator from the set, and realizes the notion of (multiple) spell-out simply as the process of lexical insertion. After Spell-out, no further grammatical dependencies can be established. It follows that the effects of quantifier raising must also be specified lexically and statically, as can be done using the generalized quantifier determiner categories (27) in section 12.4 above, rather than by post spell-out covert movement.

12.8 Conclusion

Combinatory Categorial Grammar abandons traditional notions of surface constituency in favor of "flexible" surface structure, in which most contiguous substrings of a grammatical sentence are potential constituents, complete with a monotonically composed semantic interpretation, for the purposes of the application of grammatical rules. This move achieves the following simplifications, of interest to any Minimalist Program for the theory of Grammar:

1. All covert and overt "movement" reduces to strictly type-driven (not structure-dependent) merger of string-adjacent syntactic types and logical forms projected from the lexicon by combinatory derivation.

2. The notions "phase" and "transformational cycle" reduce to the notion "lexically headed domain" at the level of logical form.

3. The notion of "spell-out" or "occasion when information is sent from syntax to phonology," whether single or multiple, is reduced to the notion of lexical insertion. Since there is no movement, there is no point in the derivation after which overt movement is prohibited, or covert movement is allowed. The relation between syntax and phonology is completely determined by the language-specific lexicon and universal combinatory projection.

4. Remaining "conditions on interface levels", such as articulatory conditions, Binding Condition C, and the island conditions are essentially external to grammar proper.

5. The availability of predominantly left branching derivations like (14) allows assembly of phonological and logical forms to be incremental or "on-line" rather than at a single stage in the computation, without any additional imposition of interface conditions, supporting dynamic approaches to Binding Conditions B and C.

Seen in this light, most remaining open questions in both CCG and MP concern the notion "possible language-specific lexical category".

In eliminating all levels of representation intervening between phonetic/phonological form and logical form, CCG is in broad accord with the more recently announced principles of the Minimalist Program, and in particular the version proposed by Epstein et al. (1998) (cf. Kitahara 1995), in which it is proposed to equate (via an unspecified mechanism) Chomsky's operations Merge and Move as a single operation. To the extent that both relativization (and other so-called movements) and in-situ argument reduction are effected in CCG by the same type-driven operation of functional application, it can be seen as providing such a mechanism. However it should be noted that in other respects the frameworks are quite different. In particular, the meaning of the term "derivation" as used by Chomsky and Epstein et al. is quite different from the sense of that term used here and in *SP*.

It is worth emphasising the point, because on occasion Chomsky has defined transformations very generally indeed, as devices "that appear to be unavoidable in one or another form, whether taken to be operations forming derivations or relations established on representations" (1995b:25). Of course, every theory of language has to define the same semantic relations for sentences at the level of logical form. However, there is a big difference between establishing those relations via a general rule of movement "freely applicable to arbitrary expressions"(ibid.), and establishing them by the equivalent of Merge over linear indexed rules or CCG. The latter is much less expressive (for example it is incapable of recognizing such simple but unnatural languages as $a^n b^n c^n d^n$ and a^{2^n}).

There are other improvements that the non-movement, type-driven, structure-independent, lexicalized interpretation of the Minimalist Program affords (cf. Chomsky 1995b:25). The role of the Numeration is now performed by the string itself, or equivalently by the notion of lexical insertion. The logical form associated with the lexical entries for heads defines the notion of Phase or local domain, and ensures that the head is projected (via the logical form) to the root node of that local domain, as required by the "bare phrase structures" illustrated in example (2). This way of formulating the computation means that linear precedence (LP) can be determined by the same grammatical module as immediate dominance (ID) or semantic dependency. The properties of weak LIG-equivalence and linearized categories together guarantee the existence of efficient performance mechanisms.

Spell-out is correspondingly simplified. It merely means that the lexicon for the language in question and the universal mechanism of combinatory pro-

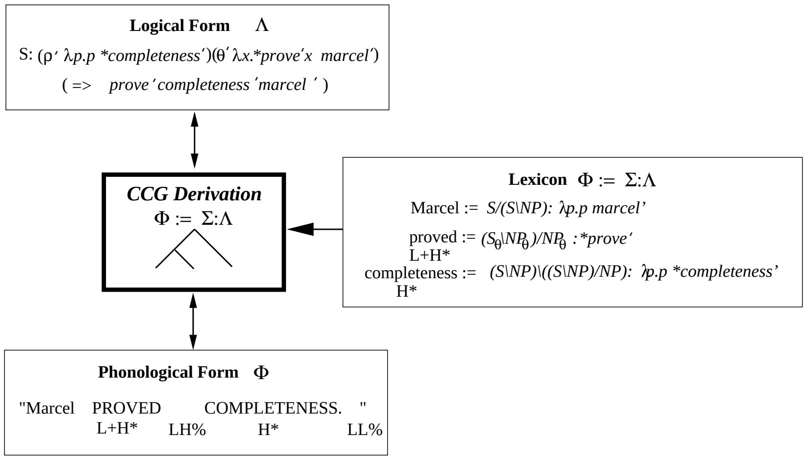


Figure 12.4
Architecture of CCG

jection support a mapping between a particular logical form and the string or PF in question. All specifics, such as whether V-raising is overt, as in French, or covert, as in English, are determined by their respective lexicons, and in particular whether the V-raising process is manifested in the lexical syntactic type of verbs, as in French *faire*, or only in the corresponding lexical logical form, as in English *make*. The fact that the equivalent of covert movement can be achieved statically in CCG, via the lexical logical form of quantifier determiners like (27) also provides a way to handle the phenomenon of quantifier raising without post- spell-out covert movement.

The architecture of the theory that results can be summarized in a version of the T- or Y- diagram standardly used in the transformationalist framework, as shown in figure 12.8. The level of PF is now a true interface level, representing only the information necessary to specify speech or orthography. The level of LF is now the sole structural level of representation, and is identified with Information Structure, a level which contains all the information that is needed for processes of verification and inference in the context of discourse. While the traditional propositional structure that is more usually associated with the notion of logical form can be trivially obtained by β-normalization of the information structure, and may be needed in order to provide a model theoretic semantics, such a representation is redundant, and not part of the theory itself.[19]

The lexicon statically assigns a triple consisting of a phonological form Φ, a syntactic type Σ, and a logical form Λ to all lexical items, and is the sole locus of language-specific information in the grammar. The combinatory rules and the process of lexical insertion map monotonically between PF and LF, also assigning a triple $\Phi := \Sigma : \Lambda$ to all elements in the derivation. These elements define a purely monotonic computation between the interface levels, of just trans-context-free automata-theoretic power, supporting standard algorithms and models for efficient application to practical tasks and realistic psychological theories. CCG thus offers not only a theory of The Computation in the sense of Marr and Chomsky, but also a way in which it can actually be practically computed.

Chapter 12

1. The Minimalist Program is a moving target, and many components are either underspecified (Merge, Phase) or are variously defined (Spell-out). All definitions are subject to change. The following summary attempts no more than to identify a consistent snapshot.

2. Other versions of combinatory categorial grammar have introduced further combinators, notably **C** (Dowty 1978, Bach 1979, Jacobson 1990).

3. Lexicalization of the syntactic component has recently been endorsed by Hale and Keyser (1993) and by Chomsky and Lasnik (Chomsky 1995b:25).

4. There is an alternative "result on top" notation due to Lambek (1958), according to which the latter category is written $\beta\backslash\alpha$.

5. These categories are deliberately simplified for expository purposes, and certainly overstate the degree to which alternative constitutent orders are semantically equivalent in these languages.

6. Baldridge (2002) shows that schematization of this kind does not increase the expressive power of the theory.

7. Fuller expositions of slash-modal CCG can be found in Baldridge (2002) and Steedman and Baldridge (2003).

8. The use of a hierarchy such as this as a formal device is optional, and instead could be replaced by multiple declarations of the combinatory rules.

9. The feature $+ANT$ can only be bound by the relative pronoun category, and excludes right node raising as in **I eat and will live to be a hundred those apricots they grow in the Caucasus*. The precise mechanism is discussed in *SP* and need not detain us here. Baldridge 2002, 109-113, shows how the same "antecedent government" feature can more elegantly be brought under the control of a slightly richer set of slash modalities, allowing these conjunctions to be simplified as versions of $(VP\backslash_{\times}VP)/_{\star}VP$ and $(VP/_{\diamond}VP)\backslash_{\star}VP$. These categories allow extraction to take place via the usual composition mechanism. They have the advantage over those in (23) of also supporting a variety of ordinary non extracted "subordinating" VP coordinations, with the same volitional causal semantics, via simple application, as in *Eat apricots and live to be a hundred!*, and *Go to the store and buy apricots!*.

10. This principle and the following Principles of Head Categorial Uniqueness and Categorial Type Transparency replace the Principle of Categorial Government in Steedman 1996.

11. An example of such a necessary exception is the treatment of subject extraction in English in Steedman 1996. It is a prediction of CCG (rather than a stipulation via a "Fixed Subject" constraint or "Empty Category Principle") that a fixed SVO word-order language like English cannot permit complement subjects to extract under the Head Categorial Uniqueness Principle, as illustrated by the anomaly of (ia). The exceptional possibility of extracting subjects from English bare complements, as in (ib), therefore

requires an extra "antecedent-governed" category for bare-complement taking verbs like *think*, in violation of that Principle.

(i) a. *Who do you think that met Monboddo?
    b. Who do you think won?

12. This question is investigated in somewhat more detail in Steedman 2005a.

13. We pass over the expletive feature *it* and the antecedent-governed feature $+ANT$, which are needed in order to prevent overgenerations like the following:

(i) a. *Monboddo seems Johnson to be happy.
    b. *It seems Monboddo to be happy.
    c. *Monboddo is likely that Johnson seems to be happy.
    d. *Monboddo is likely that it seems Johnson to be happy.

See Steedman 2005a for further discussion.

14. In more recent work, Dowty has disowned CCG in favour of TLG, because of "intrinsic" use of logical form to account for binding phenomena that it entails, as discussed above. See SSI for further discussion.

15. This appears to offer an advantage over non-type-raising accounts using the product operator $\bullet$ of Lambek (Pickering and Barry 1993; Dowty 1997).

16. It is clear that *Marcel proved* is a prosodic phrase in (53), unlike (52), because in (53), the Rhythm Rule shifts the lexical stress on *Marcel* from the last syllable to the first.

17. The much-abused term "focus" is used here strictly in the "narrow" phonological sense of the term, to refer to the effects of contrast or emphasis on a word that ensues from the presence of a pitch-accent. Elsewhere this property of accented words is called "kontrast" (Vallduví and Vilkuna 1998; Steedman 2006).

18. In a real DCG all instances of variables like $Q\hat{}\ldots$ would be instantiated at the same time with values like $\mathit{fall}'(Z)$.

19. The identification of Information Structure rather than the proposition as the representational level of logical form has also been proposed by Zubizarreta (1998).

References

Ades, Anthony, and Mark Steedman. 1982. "On the Order of Words." *Linguistics and Philosophy*, 4, 517–558.

Aho, Alfred. 1968. "Indexed Grammars—An Extension of Context-Free Grammars." *Communications of the Association for Computing Machinery*, 15, 647–671.

Ajdukiewicz, Kazimierz. 1935. "Die syntaktische Konnexität." In Storrs McCall, ed., *Polish Logic 1920-1939*, 207–231. Oxford: Oxford University Press. Translated from *Studia Philosophica*, 1, 1-27.

Asudeh, Ash, and Richard Crouch. 2002. "Coordination and parallelism in Glue Semantics: Integrating discourse cohesion and the Element Constraint." In Miriam Butt and Tracy Holloway King, eds., *Proceedings of the LFG02 Conference*. Stanford CA: CSLI Publications.

Bach, Emmon. 1976. "An Extension of Classical Transformational Grammar." In *Problems in Linguistic Metatheory: Proceedings of the 1976 Conference at Michigan State University*, 183–224. Lansing: Michigan State University.

Bach, Emmon. 1979. "Control in Montague Grammar." *Linguistic Inquiry*, 10, 513–531.

Baldridge, Jason. 1998. *Local Scrambling and Syntactic Asymmetries in Tagalog*. Master's thesis, University of Pennsylvania.

Baldridge, Jason. 2002. *Lexically Specifi ed Derivational Control in Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh.

Baldridge, Jason, and Geert-Jan Kruijff. 2003. "Multi-Modal Combinatory Categorial Grammar." In *Proceedings of 11th Annual Meeting of the European Association for Computational Linguistics*, 211–218. Budapest.

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

Berwick, Robert, and Samuel Epstein. 1995. "Computational Minimalism: the Convergence of 'Minimalist' Syntax and Categorial Grammar." In A. Nijholt, G. Scollo, and R. Steetkamp, eds., *Algebraic Methods in Language Processing 1995: Proceedings of* *the Twente Workshop on Language Technology 10, jointly held with the First Algebraic Methodology and Software Technology (AMAST) Workshop on Language Processing*. Enschede, The Netherlands: Faculty of Computer Science, Universiteit Twente.

Bolinger, Dwight. 1989. *Intonation and Its Uses*. Stanford, CA: Stanford University Press.

Bozsahin, Cem. 1998. "Deriving Predicate-Argument Structure for a Free Word Order Language." In *Proceedings of COLING-ACL '98, Montreal*, 167–173. Cambridge, MA: MIT Press.

Bozsahin, Cem. 2002. "The Combinatory Morphemic Lexicon." *Computational Linguistics*, 28, 145–186.

Brame, Michael. 1978. *Base Generated Syntax*. Seattle, WA: Noit Amrofer.

Bresnan, Joan. 1978. "A Realistic Transformational Grammar." In Morris Halle, Joan Bresnan, and George Miller, eds., *Linguistic Structure and Psychological Reality*, 1–59. Cambridge, MA: MIT Press.

Bresnan, Joan, ed. 1982. *The Mental Representation of Grammatical Relations*. Cambridge, MA: MIT Press.

Brody, Michael. 1995. *Lexico-Logical Form: A Radically Minimalist Theory*. Cambridge, MA: MIT Press.

Buszkowski, Wojciech, Witold Marciszewski, and Johan van Benthem. 1988. *Categorial Grammar*. Amsterdam: John Benjamins.

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

Chomsky, Noam. 1959. "On Certain Formal Properties of Grammars." *Information and Control*, 2, 137–167.

Chomsky, Noam. 1975. *The Logical Structure of Linguistic Theory*. Chicago: University of Chicago Press.

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

Chomsky, Noam. 1993. "A Minimalist Program for Linguistic Theory." In Kenneth Hale and Samuel Jay Keyser, eds., *The View from Building 20*, 1–52. Cambridge, MA: MIT Press.

Chomsky, Noam. 1995a. "Bare Phrase Structure." In Gert Webelhuth, ed., *Government and Binding Theory and the Minimalist Program*, 383–439. Oxford: Blackwell.

Chomsky, Noam. 1995b. *The Minimalist Program*. Cambridge, MA: MIT Press.

Chomsky, Noam. 2001. "Derivation by Phase." In Michael Kenstowicz, ed., *Ken Hale: a Life in Language*, 1–52. Cambridge MA: MIT Press.

Clark, Stephen, and James R. Curran. 2004. "Parsing the WSJ using CCG and Log-Linear Models." In *Proceedings of the 42nd Meeting of the ACL*, 104–111. Barcelona, Spain.

Clark, Stephen, Julia Hockenmaier, and Mark Steedman. 2002. "Building Deep Dependency Structures with a Wide-Coverage CCG Parser." In *Proceedings of the 40th Meeting of the ACL*, 327–334. Philadelphia, PA.

Clark, Stephen, Mark Steedman, and James R. Curran. 2004. "Object-Extraction and Question-Parsing Using CCG." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 111–118. Barcelona, Spain.

Cocke, John, and Jacob T. Schwartz. 1970. *Programming Languages and Their Compilers*. New York: New York University, Courant Institute.

Collins, Michael. 1997. "Three Generative Lexicalized Models for Statistical Parsing." In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, Madrid*, 16–23. San Francisco, CA: Morgan Kaufmann.

Cresswell, Max. 1985. *Structured Meanings*. Cambridge MA: MIT Press.

Cresswell, M.J. 1973. *Logics and Languages*. London: Methuen.

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

Dowty, David. 1978. "Governed Transformations as Lexical Rules in a Montague Grammar." *Linguistic Inquiry*, 9, 393–426.

Dowty, David. 1988. "Type-Raising, Functional Composition, and Nonconstituent Coordination." In Richard T. Oehrle, Emmon Bach, and Deirdre Wheeler, eds., *Categorial Grammars and Natural Language Structures*, 153–198. Dordrecht: Reidel.

Dowty, David. 1997. "Nonconstituent Coordination, Wrapping, and Multimodal Categorial Grammars: Syntactic Form as Logical Form." In Maria Luisa Dalla Chiara, ed., *Proceedings of the Tenth International Congress of Logic, Methodology, and Philosophy of Science, 1995*, 347–368. Amsterdam: North Holland. Extended version at `http://www.ling.ohio-state.edu/~dowty/`.

Emonds, Joseph. 1970. *Root and Structure-Preserving Transformations*. Ph.D. thesis, Massachussetts Institute of Technology.

Emonds, Joseph. 1976. *A Transformational Approach to English Syntax*. New York: Academic Press.

Epstein, Samuel D., Erich Groat, Ruriko Kawashima, and Hisatsugu Kitahara. 1998. *A Derivational Approach to Syntactic Relations*. Oxford: Oxford University Press.

Firbas, Jan. 1964. "On Defining the Theme in Functional Sentence Analysis." *Travaux Linguistiques de Prague*, 1, 267–280.

Firbas, Jan. 1966. "Non-thematic Subjects in Contemporary English." *Travaux Linguistiques de Prague*, 2, 229–236.

Gazdar, Gerald. 1981. "Unbounded Dependencies and Coordinate Structure." *Linguistic Inquiry*, 12, 155–184.

Gazdar, Gerald. 1988. "Applicability of Indexed Grammars to Natural Languages." In Uwe Reyle and Christian Rohrer, eds., *Natural Language Parsing and Linguistic Theories*, 69–94. Dordrecht: Reidel.

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

Geach, Peter. 1972. "A Program for Syntax." In Donald Davidson and Gilbert Harman, eds., *Semantics of Natural Language*, 483–497. Dordrecht: Reidel.

Goldsmith, John. 1985. "A Principled Exception to the Coordinate Structure Constraint." In *Proceedings of the 21st Regional Meeting of the Chicago Linguistic Society*. Chicago: CLS, University of Chicago.

Goodall, Grant. 1987. *Parallel Structures in Syntax*. Cambridge: Cambridge University Press.

Hale, Kenneth, and Samuel Jay Keyser. 1993. "On Argument Structure and the Lexical Expression of Syntactic Relations." In Kenneth Hale and Samuel Jay Keyser, eds., *The View from Building 20*, 53–109. Cambridge, MA: MIT Press.

Hepple, Mark. 1990. *The Grammar and Processing of Order and Dependency: A Categorial Approach*. Ph.D. thesis, University of Edinburgh.

Hockenmaier, Julia. 2003. *Data and models for statistical parsing with CCG*. Ph.D. thesis, School of Informatics, University of Edinburgh.

Hockenmaier, Julia, and Mark Steedman. 2002. "Generative Models for Statistical Parsing with Combinatory Categorial Grammar." In *Proceedings of the 40th Meeting of the ACL*, 335–342. Philadelphia, PA.

Hoffman, Beryl. 1995. *Computational Analysis of the Syntax and Interpretation of "Free" Word-Order in Turkish*. Ph.D. thesis, University of Pennsylvania. IRCS Report 95-17.

Jacobson, Pauline. 1990. "Raising as Function Composition." *Linguistics and Philosophy*, 13, 423–476.

Jacobson, Pauline. 1992. "Flexible Categorial Grammars: Questions and Prospects." In Robert Levine, ed., *Formal Grammar*, 129–167. Oxford: Oxford University Press.

Joshi, Aravind, Leon Levy, and M. Takahashi. 1975. "Tree-adjunct Grammars." *Journal of Computer Systems Science*, 10, 136–163.

Joshi, Aravind, K. Vijay-Shanker, and David Weir. 1991. "The Convergence of Mildly Context-Sensitive Formalisms." In Peter Sells, Stuart Shieber, and Tom Wasow, eds., *Processing of Linguistic Structure*, 31–81. Cambridge, MA: MIT Press.

Kayne, Richard. 1994. *The Antisymmetry of Syntax*. Cambridge, MA: MIT Press.

Kehler, Andrew. 2002. *Coherence, Reference, and the Theory of Grammar*. Stanford CA: CSLI Publications.

Kimball, John. 1967. "Predicates Definable over Transformational Derivations by Intersection with Regular Languages." *Information and Control*, 11, 177–195.

Kitahara, Hisatsugu. 1995. "Target α: Deducing Strict Cyclicity from Principles of Derivational Economy." *Linguistic Inquiry*, 26, 47–77.

Komagata, Nobo. 1999. *Information Structure in Texts: A Computational Analysis of Contextual Appropriateness in English and Japanese*. Ph.D. thesis, University of Pennsylvania.

Koster, Jan. 1978. "Conditions, Empty Nodes, and Markedness." *Linguistic Inquiry*, 9, 551–593.

Kuno, Susumu. 1965. "A System for Transformational Analysis." In *Proceedings of the International Conference on Computational Linguistics (CoLing)*. ACL Anthology.

Lakoff, George. 1986. "Frame Semantic Control of the Coordinate Structure Constraint." In Anne Farley, Peter Farley, and Karl-Eric McCullough, eds., *Papers from the Parasession on Pragmatics and Grammatical Theory at the 22nd Regional Meeting of the Chicago Linguistics Society*, 152–167. Chicago: CLS.

Lambek, Joachim. 1958. "The Mathematics of Sentence Structure." *American Mathematical Monthly*, 65, 154–170.

Marr, David. 1977. "Artificial Intelligence: A Personal View." *Artifi cial Intelligence*, 9, 37–48.

Mathesius, Vilem. 1929. "Functional Linguistics." In Josef Vachek, ed., *Praguiana: Some Basic and Less Well-known Aspects of the Prague Linguistics School*, 121–142. Amsterdam: John Benjamins. Publ. 1983.

Moortgat, Michael. 1988. *Categorial Investigations*. Ph.D. thesis, Universiteit van Amsterdam. Published by Foris, Dordrecht, 1989.

Morrill, Glyn. 1994. *Type-Logical Grammar*. Dordrecht: Kluwer.

Oehrle, Richard. 1975. *The Grammatical Status of the English Dative Alternation*. Ph.D. thesis, MIT.

Oehrle, Richard T. 2000. "Multi-modal Type Logical Grammar." In Robert Borsley and Kersti Borjars, eds., *Non-Transformational Syntax: A Guide To Current Models*, in preparation. Oxford: Blackwell.

Oehrle, Richard T., Emmon Bach, and Deirdre Wheeler, eds. 1988. *Categorial Grammars and Natural Language Structures*. Dordrecht: Reidel.

Ojeda, Almerindo. 1988. "A Linear Precedence Account of Cross-Serial Dependencies." *Linguistics and Philosophy*, 11, 457–492.

Partee, Barbara, and Mats Rooth. 1983. "Generalised Conjunction and Type Ambiguity." In Rainer Baüerle, Christoph Schwarze, and Arnim von Stechow, eds., *Meaning, Use, and Interpretation of Language*, 361–383. Berlin: de Gruyter.

Peters, Stanley, and Robert Ritchie. 1973. "On the Generative Power of Transformational Grammars." *Information Science*, 6, 49–83.

Pickering, Martin, and Guy Barry. 1993. "Dependency Categorial Grammar and Coordination." *Linguistics*, 31, 855–902.

Pierrehumbert, Janet. 1980. *The Phonology and Phonetics of English Intonation*. Ph.D. thesis, MIT. Distributed by Indiana University Linguistics Club, Bloomington.

Pierrehumbert, Janet, and Mary Beckman. 1988. *Japanese Tone Structure*. Cambridge, MA: MIT Press.

Pollard, Carl, and Ivan Sag. 1994. *Head Driven Phrase Structure Grammar*. Stanford, CA: CSLI Publications.

Postal, Paul. 1998. *Three Investigations of Extraction*. Cambridge, MA: MIT Press.

Prevost, Scott. 1995. *A Semantics of Contrast and Information Structure for Specifying Intonation in Spoken Language Generation*. Ph.D. thesis, University of Pennsylvania.

Prevost, Scott, and Mark Steedman. 1994. "Specifying Intonation from Context for Speech Synthesis." *Speech Communication*, 15, 139–153.

Rooth, Mats. 1985. *Association with Focus*. Ph.D. thesis, University of Massachusetts, Amherst.

Rooth, Mats. 1992. "A Theory of Focus Interpretation." *Natural Language Semantics*, 1, 75–116.

Ross, John Robert. 1967. *Constraints on Variables in Syntax*. Ph.D. thesis, MIT. Published as *Infi nite Syntax!*, Ablex, Norton, NJ, 1986.

Shieber, Stuart. 1985. "Evidence against the Context-Freeness of Natural Language." *Linguistics and Philosophy*, 8, 333–343.

von Stechow, Arnim. 1991. "Focusing and Backgrounding Operators." In Werner Abraham, ed., *Discourse Particles: Descriptive and Theoretical Investigations on the Logical, Syntactic and Pragmatic Properties of Discourse Particles in German*, 37–84. Amsterdam: John Benjamins.

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

Steedman, Mark. 1996. *Surface Structure and Interpretation*. Cambridge, MA: MIT Press.

Steedman, Mark. 2000a. "Information Structure and the Syntax-Phonology Interface." *Linguistic Inquiry*, 34, 649–689.

Steedman, Mark. 2000b. *The Syntactic Process*. Cambridge, MA: MIT Press.

Steedman, Mark. 2005a. "Interfaces and the Grammar." In *Proceedings of the 24th West Coast Conference on Formal Linguistics, Vancouver, March 2005*, 19–33. Cascadilla Proceedings Project.

Steedman, Mark. 2005b. "Surface-Compositional Scope-Alternation Without Existential Quantifiers." URL `http://www.cogsci.ed.ac.uk/~steedman/papers.html`. Draft 5.1. University of Edinburgh, Sept 2005.

Steedman, Mark. 2006. "Semantics and Implicature in the Meaning of English Intonation." *Submitted*.

Steedman, Mark, and Jason Baldridge. 2003. "Combinatory Categorial Grammar." In Kirsti Börjars, ed., *Non-Transformational Syntax: A Guide to Current Models*. Blackwell. To appear, maybe.

Steedman, Mark, and Jason Baldridge. 2006. "Combinatory Categorial Grammar." In Keith Brown, ed., *Encyclopedia of Language and Linguistics*, vol. 2, 610–622. Oxford: Elsevier, second edn.

Svenonius, Peter. 2004. "On the Edge." In David Adger, Cécile de Cat, and George Tsoulas, eds., *Peripheries*. Dordrecht: Kluwer.

Thorne, James, Paul Bratley, and Hamish Dewar. 1968. "The Syntactic Analysis of English by Machine." In Donald Michie, ed., *Machine Intelligence, Vol. 3*. New York: American Elsevier.

Trechsel, Frank. 2000. "A CCG Account of Tzotzil Pied Piping." *Natural Language and Linguistic Theory*, 18, 611–663.

Vallduví, Enric, and Maria Vilkuna. 1998. "On Rheme and Kontrast." In Peter Culicover and Louise McNally, eds., *Syntax and Semantics, Vol. 29: The Limits of Syntax*, 79–108. San Diego, CA: Academic Press.

Vijay-Shanker, K., and David Weir. 1990. "Polynomial Time Parsing of Combinatory Categorial Grammars." In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics, Pittsburgh*, 1–8. San Francisco, CA: Morgan Kaufmann.

Vijay-Shanker, K., and David Weir. 1993. "Parsing Some Constrained Grammar Formalisms." *Computational Linguistics*, 19, 591–636.

Vijay-Shanker, K., and David Weir. 1994. "The Equivalence of Four Extensions of Context-Free Grammar." *Mathematical Systems Theory*, 27, 511–546.

Wood, Mary McGee. 1993. *Categorial Grammar*. London: Routledge.

Woods, William. 1970. "Transition Network Grammars for Natural Language Analysis." *Communications of the Association for Computing Machinery*, 18, 264–274.

Woods, William. 1973. "An Experimental Parsing System for Transition Network Grammars." In Randall Rustin, ed., *Natural Language Processing*, 111–154. New York: Algorithmics Press.

Zubizarreta, Maria Luisa. 1998. *Prosody, Focus, and Word Order*. Cambridge, MA: MIT Press.