Linearisation: adjuncts and arguments

ANNABEL CORMACK AND NEIL SMITH

Abstract

Kayne (1994) introduced a 'Linear Correspondence Axiom' which offered for the first time a principled theory concerning the linear PF ordering of NL phrases and words. Within this theory, adjuncts constitute an anomaly, requiring a stipulative definition of c-command. If the category labels of Categorial Grammar are used, a simpler linearisation depending on Asymmetric Merge is obtained, which applies without stipulation to adjuncts. The linearisation makes some distinct and desirable predictions.

1 Introduction

This paper is part of a larger enterprise to develop a Minimalist version of Categorial Grammar (MCG). Here, we discuss the basis of word-order in Natural Language and propose an alternative to theories based on Kayne (1994). The new Merge-based alternative makes distinct empirical predictions, a few of which we will discuss. Our immediate purpose is to argue for a simple linearisation algorithm within the MCG we have been pursuing, but we think it could be adopted within a more standard Minimalist grammar, with simplification to the system.

We assume a grammar that pairs PF and LF representations which are interpreted at the Sensory-Motor and Intentional-Conceptual interfaces respectively. These interpretations result in sounds or gestures unfolding over time and/or space on the one hand, and in contextually mediated effects on mental representations on the other. The apparently unlimited expressive possibilities offered by Natural Language are to a large extent due to the combinatorial options made usable by the assumption of compositionality: new LFs may be constructed by using one or more Merge options, each of which takes as input a pair of well-formed LF-PF pairs (<LF1, PF1>, <LF2, PF2>), and outputs a new larger LF-PF pair <LF3, PF3> whose parts are interpretable at the interfaces by virtue of the properties of the Merge and the content of the input pairings.

The term 'displacement' is used throughout to refer to some mechanism by which an item occurs in a position not predicted by the simplest version of the grammar. In Minimalism, this includes items displaced by 'Internal Merge', leaving copies in the earlier position. In the Combinatory Minimalist Grammar we have been exploring, 'displacement' refers only to what falls under the 'Split Signs' hypothesis, under which it is possible for the PF-part of some sign to be merged in a position distinct from its LF-part, and where the split is driven by morphological requirements.[1] All other configurations are produced directly by ('external') merge. The driving force for merge is the LF component of the lexicon, with PF being parasitic (Cormack and Smith 1999, 2000 a, b).[2]

2 Linearising PF

2.1 Theories of linearisation

In current Minimalist theory, syntactic structures are constructed recursively through the application of binary merge, where the items merged may be heads or phrases. This induces a tree structure, where PF requires a temporal ordering on the set of leaves associated with information to be interpreted at the A-P interface. This linearisation of a set of items requires two components: an ordering relation on the items, and a mapping from that ordering to the temporal ordering. A linear ordering relation must be **asymmetric**, **irreflexive**, **transitive** and **total**.[3] Because of the unbounded nature of linguistic phrases, the ordering must be based on some recursive relation.

In many grammars, the basis for the PF order is given by lexical stipulation (which will include generalisations over the lexicon). Examples are HPSG, most Categorial Grammars, and X-bar theory. Under the latter, the recursive component analyses branching nodes as consisting of either $[_{X'}$ X YP] or $[_{XP}$ X′ ZP]; for each, a mapping parameter determines which of the constituents comes first. The induced ordering is too crude to give the correct NL linearisation, but since X-bar-theory was placed within a grammar with 'movement', any superficial failure of a language to reflect the general ordering principle could be ascribed to movement: that is, to the displacement licensed or required in different languages. Kayne's 1994 innovation was to propose that linearisation is uniform for all languages, and is not based on the lexicon, but essentially on the geometric structure of the tree. The ordering relation is based on asymmetric c-command. Chomsky (1995 a, b) adapted Kayne's theory to the Minimalist program. We discuss some disadvantages of the resulting theory below.

[1] We expect the resulting 'PF-displacement' to be: (i) for output reasons (e.g. fewer word boundaries, heaviness and memory limitations); (ii) to make life easier for the listener (focus marking, shape conservation, adverbial roll-up); or (iii) left over from a different lexicon (from language change). We are just assuming this, here (see also Chomsky 2004: 3).

[2] See also Chomsky (2004: 3) regarding the asymmetry of the interfaces.

[3] A relation R is asymmetric iff for all a,b,c aRb $\Rightarrow \neg$ bRa; it is irreflexive iff for all a, $\neg$ aRa; it is transitive iff all a, b, c, aRb $\wedge$ bRc $\Rightarrow$ aRc; and it is total if for all a, b, either aRb or bRa (where a,b,c are in the domain of R throughout).

Our own proposal, couched within a combinatorial version of minimalism, is algebraic rather than geometric. Instead of being based on tree-structure, it is based on an asymmetry in Merge itself. Essentially, this allows us to see the 'Spec-head' and 'head-complement' relations of X-bar theory as instances of a single relation, recursively supplying an ordering of sister nodes on the tree.[4] A single parameter allows for two alternative mappings to the temporal order. The difference between the base ordering and the temporal ordering can be seen by analogy: consider a train journey to visit a set of towns, {Edinburgh, York, London, Newcastle}. The set of towns is 'linearised' by geographical and monetary economy, so that York lies between London and Edinburgh, and Newcastle between York and Edinburgh. We can write this ordering as <Edinburgh, Newcastle, York, London>. But a temporally realised train journey requires a parametric choice in addition: start at the North (Edinburgh) or start at the South (London).

2.2 Kayne's Linear Correspondence Axiom in the Minimalist program

Kayne's 'Linear Correspondence Axiom' (LCA), derives a linear ordering *d* on the set of terminals *T* of a phrase (1994: 5-6). If *d* is set to map to 'precedence', then we obtain the axiom given in (1). Kayne argues that there is no parametric variant of the mapping given, (1) itself is also generally referred to as the LCA.

(1) Let *X*, *Y* be non-terminals, and *x*, *y* terminals such that *X* dominates *x* and *Y* dominates *y*. Then if *X* asymmetrically c-commands *Y*, *x* precedes *y*.
(Kayne 1994: 33)

This works smoothly for a purely right-branching structure, such as that in (2):

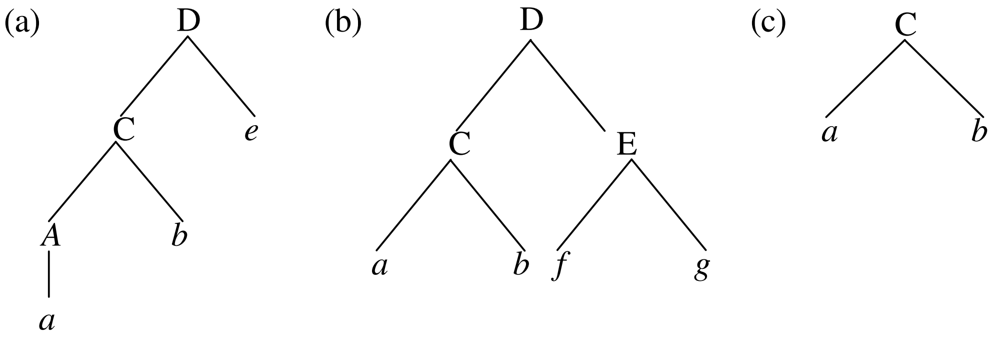
(2) ….. [said [that [Bill [might [think [that [Susie [must [never [….

C-command over the non-terminals above these items is asymmetric. Hence the lexical items can be linearised by the LCA, as has been done in (2) above.

Unfortunately for this simple idea, NL has complex adjuncts and specifiers, which disrupt the simple application of asymmetric c-command to the general tree, and there is a problem at the terminus of the tree. Consider the sort of simple tree shown in (3a), in which dominance is to be read as significant, but not the ordering of sisters. The task of a linearisation axiom is to produce an ordering of the leaves of the tree.

[4] As required by a minimalist stance: cf. Chomsky (2005: 11), suggesting that the distinction is unnecessary, reducing to first vs. later merge.

(3)



Abstracting away from labels, and using asymmetric c-command of the non-terminals as a PF temporal ordering requirement, we will in (3a) obtain $e < a$, and $e < b$. It follows that $e <$ C, although this is not derivable directly. However, if in (3b) the leaves are terminals, where E is the mother of *f* and *g*, then no non-terminal asymmetrically c-commands any other, so no linearisation can be obtained. Nor will asymmetric c-command order *a* with respect to *b* in the sub-case of (3b) given in (3c). Kayne includes under the LCA a subset of trees like (3b), those where C=D or E=D, by appealing to categories with 'two segments' (Chomsky 1986: 7; deriving from May 1985: 34); here C will be either an adjunct or a specifier. For this to work, C-command must be elaborated as in (4):

(4) *X* c-commands *Y* iff *X and Y are categories* and X *excludes* Y and every category that dominates *X* dominates *Y*. (*X* excludes *Y* if no segment of *X* dominates *Y*)

Kayne then took the LCA as a filter, discarding as impossible to order, trees like the residual instances of those in (3b), and those in (3c).

Chomsky's (1995b) adapts Kayne's proposals to bare phrase structure. Under bare phrase structure, there is no terminal vs. non-terminal distinction, so there is no separate node A in (3a). Chomsky's version of the LCA had four parts. Three eliminate problems for particular structures: the merge of a pair of terminals; specifiers; and adjuncts. The fourth, taken over from Kayne, throws out any remaining structures that cannot be linearised: the LCA still acts as a filter. The filter function is consistent with the minimalist program's rationale, since it imposes a requirement (for linearisation) that is naturally required by the A-P interface, but is inimical to the (perhaps unattainable) desideratum of 'crash proof syntax' (Frampton and Gutmann 2002). The three rescue devices each have problems.

Chomsky (1995a: 334 ff) allows situations like (3c), but only if one of the items merged is either an affix, or empty, at PF, and so needs no ordering relative to the other.[5] It follows that if V *see* is merged with the bare noun phrase *Mary*, *Mary* must move, for instance. Since there must be a syntactic feature driving this movement, it will be required in general that the object of a head X move out of the XP. Huge amounts of movement must take place.

Specifiers were assimilated to adjuncts by Kayne, but are treated differently by Chomsky. For the specifier case, Chomsky reinstates bar levels under a different guise, (1995a: 339 and 1995b: 83)). For the adjunct cases, appeal is made as in Kayne (1994) to categories with ‘two segments’, with the accompanying c-command kludge.

More crucially, c-command itself is probably not the fundamental relation it is assumed to be (cf. Chomsky 2004: 8). C-command within a carefully chosen structure like that in (2) corresponds to the logical notion of scope, so that it makes sense at an intuitive level that c-command can be replaced by a combination of semantic scope and selection.

Under the Minimalist Program, the primary relation given by the grammar is Merge; hence, in our attempt to avoid these problems, we will consider linearisations based on the asymmetry of Merge.

3 The asymmetry of Merge

3.1 Projection, selection, and the asymmetric basis for ordering

Reliance on iterable Merge as the sole computational operation of narrow syntax (Chomsky 2004: 6)

Our proposal is consistent with the rationale both of the Minimalist program and of Categorial grammars, in that it is based on an inherent asymmetry of (binary) Merge. The ordering requirement we will impose is a generalisation of the ‘head parameter’ of X-bar theory to a ‘Functor parameter’, where a functor is the representation of an item behaving locally as a function under merge: and the functor is to precede/follow the representation of its argument at Merge.

The idea underlying Merge as the basis for linearisation is that in order to avoid problematic cases like those of the last section, it is sufficient and non-redundant to linearise the daughters of every node. For example, the trees in (3) have been arbitrarily linearised on the page by setting C<E/e, *a*<*b*, and *f*<*g*, where ‘<’ represents ‘to the left of’. The linearisation will then be induced as the items are

[5] The affix-head ordering will be given in the morphological component.

merged — the extreme of cyclic spell out.[6] The question then is what asymmetric property of Merge, if any, can serve as a suitable basis for linearisation.

In bare phrase structure, when two items with labels A and B are merged, just one of them projects its label (Chomsky 2000: 133-4). Here we seem to have all the ingredients of an asymmetric relation on Merge. But the details of standard merge are unsatisfactory. We have either 'set merge' or 'pair merge' corresponding to the older notions of 'substitution' and 'adjunction' respectively. Set merge is inherently asymmetric: if a head A **selects** a complement B, then A has a selection feature which is the probe driving the merge. It is the item having the selection feature that projects its label (this being the only one that has a feature always required in the merge (ibid: 134). In pair merge, if A is an **adjunct**, and B is its **host**, then B projects its label. But the adjunct-host projection is entirely stipulative. Further, if as argued by many (e.g. Cormack and Breheny 1994, Sportiche 1994, 1995), adjuncts are also taken to select, we should not need to distinguish 'pair merge' from 'set merge'. It should then be the adjunct whose 'label' projects; but under conventional labelling, this would fail to represent the optionality of adjuncts correctly (and adverb-verb merge would give rise to an adverb projection, instead of a verb projection).

We take the specific proposals of a Categorial Grammar lexicon to elucidate and resolve these issues. Once we have done that, we see that the asymmetry of projection at merge does after all provide a satisfactory basis for linearisation.

The basic tenet of a CG is that syntax and semantics should and can be given the same algebraic structure. If semantics is modelled using recursive steps of function-argument application, then syntactic categories should also be modelled by function-argument application. Hence if we take the meaning of *John sleeps* to be obtained by applying the function **sleeps′** of semantic type (e → t) to **john′** of type (e), yielding a result of type (t), then we should also take an intransitive verb to be a function from category D to category V, applying to the noun of categorial type D, and yielding a result of category V. Hence, an intransitive verb *sleep* has category (D → V). For the categorial type, we will use the slash notation, V/D, though we first introduce the [/D] as a selection feature; and for the semantic type we use the parallel notation, <t/e>.

In every merge of two items A and B, some features of A and of B must or may project to the mother; while other features of A and of B might not project.[7] We assume the 'inclusiveness' condition (Chomsky 1995a: 228), and we also take it that merge is only licensed if at least one feature is discharged (Chomsky 2000:

[6] "Phases should, presumably, be as small as possible, to minimize computation after Transfer and to capture as fully as possible the cyclic/compositional character of mappings to the interface." (Chomsky 2004: 21).

[7] We use 'item' to refer to any head or well-formed phrase.

133, 2005: 6). We will also assume that the algebra of merge is the **only** means by which selection features can be checked.[8] What kind of features are suitable to give the required asymmetry? The kind of feature required here must be a categorial feature or a type feature, since only these are necessarily associated with every item. That heads and phrases may have selection features is a consequence of the arity (valency) of the head, encoded in syntax as s-selection.[9] When a phrase has zero arity, it no longer induces merge. For familiarity, we generally record only categorial selection features in the discussion below, although type features are equally important. All instances of Merge then must satisfy (5):

(5) Merge is licensed by the discharge of a selection feature

Let us use a feature with a preceding slash to denote a selection feature. A selection feature [/α] is discharged when it is in a sisterhood relation with some item with the feature [α]; further, the matched feature [α] is also discharged. Consider a verb like *think*, which selects for both an entity (the thinker) and a proposition (the thought). The verb, then, may be seen as requiring an external argument, an entity of type <e>, and complement, a proposition, of type <t>. Such types give the s-selection properties of a head. Further, when the head is saturated by the discharge of its arguments, the result is a proposition. We will take it as given here that NL also requires that heads are given categories, so that arguments are additionally specified in terms of c-selection.[10] For *think*, the canonic c-selections are respectively for DP and CP.[11]

We can give the merge for the two kinds of features as in (6a) and (6b), which represent the merge of two items to construct a third, the mother. The (semantic) types are given in reversed, non-standard format, so that the order of input and output is parallel to that of the (syntactic) categories. Generally, for categories we

[8] That is, there is feature-percolation, rather than agreement at a distance.

[9] Nevertheless, we allow that there might in a minority of cases be a discrepancy between the arity of a conceptual, Language of Thought (LoT) item, and the NL item which relates to this (for example, for unaccusatives, which might be simple one-place predicates in LoT).

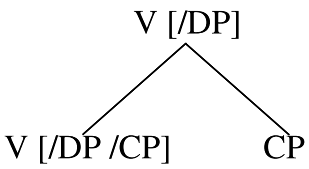
[10] That c-selection is real, even if often predictable within a given language, is evident from contrasts not due to types. For example, *likely* and *probable* arguably have the same meaning, but differ in that one allows Raising where the other does not: *John is likely/*probable to win.* Another example is the choice of coding as adjective (English) or verb (Nupe) of a concept like RED, a one-place predicate (contrast Kayne, 2005: 44 ff). That humans divide predicate words into nouns or verbs or adjectives is no more surprising than the fact that they innately divide objects into animate and inanimate (cf. Kihm 2005; 460-461 "the spontaneous and irrepressible need we have of pigeonholing all the things in the world …"; words are objects in the world too).

[11] We abstract away here from the possibility that the external argument is introduced by an intermediate head like 'little v'; the extra structure can be incorporated without any relevant problems if there is independent justification for it.

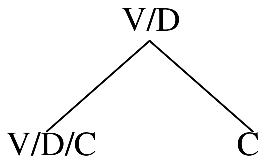
will be using the bare abbreviated notation, with the feature-brackets omitted, like that in (6c). From now on, labels such as DP, CP etc. are to be read as informal ways of referring to saturated projections of D, C etc.; and for lexical X, XP is to refer, as is familiar, to the one-place predicate, X/D in the bare notation. An argument noun-phrase, which is not saturated, may be referred to informally as DNP or DetNP. We have also assumed that the argument selections are not given as a set, but as a list, to be discharged in succession (that is, if the head is considered as an n-place operator, it has been Curried, which induces multiple binary branching).[12]

(6)	a.	*item1*:	V [/DP /CP];	type <t/e/t>
		item2:	CP;	type <t>
	Merge *items 1* and *2* to give	*item3*:	V [/DP]	type <t/e>

b.

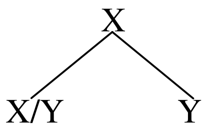


c.



We now have sufficient information in the feature structure to give an asymmetric merge relation here. We will discuss selection in terms of c-selection, rather than s-selection, since this gives a more informative label which is easier to process. The categorial feature [V] of *item1* is percolated to the mother, but not that of *item2* [CP]. A categorial selection feature [/CP] of *item1* is discharged (under matching with the categorial feature of item2) during this merge, but not one on *item2*. The general case is as in (7), where X and Y are underdetermined categories.

(7)



On the basis of this, it looks as though we could use either of the two asymmetries (categorial selection projection, or categorial feature discharge) as a basis for ordering. If one of these is the correct ordering principle, then it should extend without special provision to adjuncts and noun-phrases including specifiers. The next sections consider these.

[12] The ordering might be given by UTAH (Baker 1988), or by other mapping principles.

3.2 Adjuncts as operators

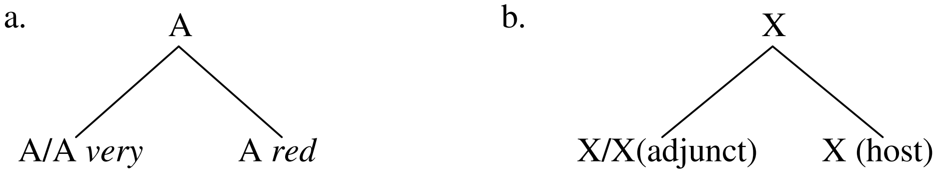
It is widely recognised that the current standard Minimalist theory of adjuncts and modifiers is unsatisfactory. Wilder and Gärtner (1997: 18) go so far as to say, "The issue of adjunction stands out as defying full integration with minimalist principles. It is an open question whether the cost incurred by the technical complications which adjunction necessitates might not actually make it desirable to dispense with adjunction altogether." Sportiche (1995: §4.2 likewise argues for eliminating adjuncts. We agree with both the diagnosis and the elimination suggested.

In many, but by no means all, grammars, the adjunct–host relation is also seen as mediated by selection, so that the adjunction structure conforms to (5) above. This principle has always been a feature of Categorial Grammars, and can unproblematically be adopted in P&P based grammars.

In a standard categorial grammar, selection features are considered to be part of the category, and are simply marked off by slashes, as in the ‘V/D/C’ notation in (6c). We will use this simpler notation, and come in a moment to some motivation for considering the features to be part of the category. The notation can be read as stipulating for any item, the category of any items with which it must combine (the first selection it makes), and the category that results from merging an item with a selected complement.[13] The general case shown is in (7). Provided that the selection features are considered to be part of the category label, X and Y need not be simple categories: X is not necessarily a head.[14] Hence X might be U/V, so that the functor category expands as (U/V)/Y; this is normally written as U/V/Y.

It now becomes straightforward to assign a complex category label to an adjunct like *very*. If the item selects for an adjective and returns the category adjective after merge, then we must put Y=A and X=A in the tree in (7), to yield (8). More generally, we have the categories and tree in (8b) for a simple head which is adjoined to a category X.

(8)



[13] In a standard categorial grammar (see Steedman 2000, chapter 3), the direction in which the slash leans also encodes linearisation information. This additional interpretation is not to be imposed on the slashes in our usage.

[14] Everything that is said about categories applies equally to the types. In particular, an adjunct does not change the type of its host.

That is, an adjunct item is a one-place operator, with category X/X for some X, where X is the category of the host of the adjunct.[15] Note now that the category of the mother is the base category of each of the daughters, so it is not obvious that it makes sense to ask which category has projected to give the label of the mother. However, a selection feature of just one daughter (the adjunct) has been discharged, under matching with the category of the host, so this asymmetry can be used as a basis for ordering. Simple adjuncts conform as required to the ordering relation dependent on selection-feature discharge. We therefore take this as the basis of the ordering relation.

Let us refer to the item whose accessible (outermost) selection feature is discharged as the 'functor' in the merge relation; the other item will be the 'argument' in the merge relation. This assumption allows us to recursively specify linearisation via binary merge.

We assume, as in a simple categorial grammar, that the interpretation at LF is derived by function-argument application. The inductive step in constructing a complex structure is thus as in (9).

(9) Two items may be merged if their categories may be analysed as X/Y and Y (X, Y not necessarily simple categories). If the PF content of X/Y is α, and the PF content of Y is β, then the PF γ of the mother X is given by a linearisation of α and β:

PF-ordering of γ in X: $[_{X/Y}\ \alpha] < [_{Y}\ \beta]$

Linearisation parameter: '<' maps onto 'temporally precedes'
'Functor First'

or '<' maps onto 'temporally follows'
'Functor Last'

The linearisation parameter is probably a property of the Sensory-Motor interface, rather than of the PF component of the grammar (as Chomsky 1995a : 340 suggests of the LCA).

At this point, what we have approximates to a categorial grammar interpretation of X-bar theory, although since there is just one linearising parameter rather than two (a head parameter and a specifier parameter), we might expect an even poorer

[15] Here again, X might in principle not be a simple category. For example, an adjunct to a VP would need to have category (V/D)/(V/D), unless function composition or type lifting is available for the merge (as in a standard categorial grammar), in which case it may have category V/V.

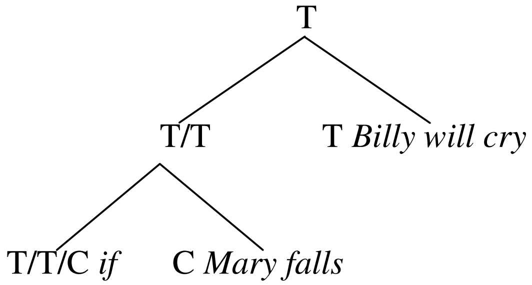
fit to an NL than X-bar theory without a displacement component. Unlike a standard categorial grammar, too, we only have one slash-direction, so that again, it might seem that a poor fit would be obtained. However, we will see in the next section that when two-place operators and binders are taken into account, the ordering defined so far, together with some 'displacement' for verbal heads, predicts the basics of ordering for all NLs more adequately than the alternatives.

4 Complex adjuncts and specifiers

4.1 Two-place operators

To simplify the discussion in these sections, we will assume that the English linearisation parameter has been set: Functor First. If the adjunct is a single lexical item, its complex category X/X will be given in the lexicon. If the item is complex, it will acquire its category after discharge of other complements (properly, OPERANDS). For example, the head *if* may have the category T/T/C: that is, it may select initially for a CP (with null C), and then for a TP, yielding a modified TP. The Functor First linearisation rule will induce ordered trees of the form in (10):

(10)



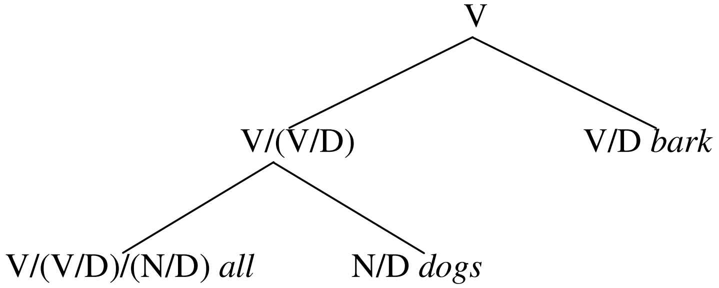
Note in particular the linearisation induced at the intermediate level, because of the functor node T/T. Then for initial complex adjuncts, nothing further needs to be said, and the linearisation hypothesis gives the desired result. The alternative *Billy will cry if Mary falls* may be derived by merging *Billy will cry* in initial position (licensed via **Phon,** or **Gap**, as in Cormack and Smith 2000b), or by low adjunction of *if Mary falls* (following Stroik 1990; the semantics is given in Cormack and Smith 2005: 415-6).

There are some adjuncts for which there is not an obvious selecting head. For example, an AP (i.e. A/D) does not select for an NP (N/D), but can modify one, as in *round plate*. This and other such cases fall under adjunction mediated by a covert asymmetric conjunction head (Cormack and Smith 2005).

4.2 Noun phrases

We argue next that there is no need for a notion ‘specifier’ to accommodate argument noun phrases (Cormack 1999, and within the standard Minimalist Program, see Starke 2001). As in previous work, we take it that all noun-phrases are headed by a quantifier (including definite and indefinite determiners; for proper nouns and pronouns, see below). We give an unsaturated argument noun phrase the informal label DNP. A quantifier is a head which semantically binds the open argument selection in two one-place predicates, the NP ‘restrictor’ and the XP ‘nuclear scope’ or ‘body’ of the operator. That is, semantically, a determiner heads a two-operand tree, rather as *if* does. Consider the determiner which heads a subject, where the predicate is an intransitive item. The restrictor is the first operand of the determiner and is a one-place predicate, with category N/D, where D is the category for a simple referring expression.[16] The second operand is a lexical predicate, say some projection of V (ignoring T for the moment), so that its full category will be V/D. Thus we expect the tree to be as in (11):

(11)



The result is a projection of V which is fully saturated: there are no selection features left. Our serialisation rule gives us the order above. In particular, at the final merge, the V/(V/D) must be the item corresponding to X/Y in (7), with V/D corresponding to Y.

Determiner headed phrases (DNPs) which are internal arguments need a modified category and type: we assume the required systematic variation in selection properties of DNPs.[17] The general category for an argument DNP is that in (12), where X is an underdetermined category, which need not be simple:

(12) **X**/(**X**/D)/(N/D) (where X need not be saturated)

[16] We follow the Montagovian position that as a matter of fact, NL supplies no such simple items, with the possible exception of clitic pronouns.

[17] See discussion and references in Heim and Kratzer (1998: § 7.2.1).

For example, object DNPs will require X= V/D, where for subject noun phrases, we had X=V. These further forms also give rise to trees where the serialisation rule puts the determiner first, followed by the NP, and then the predicate which is bound, as was the case in (11).

It is assumed that proper nouns and pronouns (perhaps excepting clitic pronouns) are 'nouns'. Thus they cannot satisfy a selection for D. A proper name has category N and type <e>. In an argument noun-phrase, it must be preceded by a type-lifting null determiner, notated as '↑':

(13) ↑ has category: X/(X/D)/N; type <t/<t/e>/e>;
semantics: $\lambda x\ \lambda P\ Px$ for x of type <e> and P of type <t/e>

The associated noun-phrase will now have category X/(X/D), so that it will precede the lexical head or phrase as usual.

5 Predictions so far for clause structure

5.1 Basic order

With the Categorial Grammar description of an adjunct, and with a semantically-based notion of a DNP, we can use the essential asymmetry of Merge to order sisters, dispensing with all the baroque definitions relating to 'segments' of categories etc., together with the notion 'specifier' in relation to arguments.

The Functor First rule gives an initial linearisation based on merge. In this respect, it is more like the older 'head first' parameter setting than like Kayne's proposals, which linearise for PF purposes an unordered tree which may already have been subject to movement. However, 'Functor First' makes predictions which differ from those given by a 'head first' parameter setting, or by an Asymmetric C-command based account. Adjuncts are initial in their merge, without special provision. In a simple clause, where there is no 'displacement', a lexical head will precede its arguments, except when the selection is discharged by a binding DNP, when the DNP will precede the head. Thus the order predicted for a verb of category V/D/D/X and its arguments (where DNP is a binding determiner-headed phrase, and X is not DNP) is as (14):

(14) $\mathrm{DNP}_{\mathrm{subject}}\ \mathrm{DNP}_{\mathrm{object}}$ V X

This ordering is neither 'head initial' nor 'head-final' as usually understood, so that the implications of all arguments that have been put forward for one or the other need to be rethought in the light of this further option. This ordering, SOVX, with clausal arguments following the verb, occurs frequently in NLs. Displacement of V would derive SVOX, another common order (more on this below), or VSOX. If children use the temporal relation of subject and object to the verb to establish the linearisation parameter for their language (a hypothesis which is not obviously true), then it appears that the vast majority of the world's languages have a 'Functor First' setting. Languages with O preceding S are rare (according to Tomlin, cited by Hawkins (1994), only around 4% of languages are like this). Nevertheless, we cannot envisage any non-arbitrary mapping from the functor–argument asymmetry of merge to temporal ordering, so that we take the linearisation parameter to embody a real choice available to NLs.

In the next few subsections, we spell out some desirable consequences of the new LF order.

5.2 Quantifier scope

One considerable advantage of the Functor First analysis is that it predicts immediately and correctly that in every such language, the 'natural' scope order of quantifiers is the same as the linear order in which they are produced. It is inverse scope, where this is allowed in languages, that has to have special treatment. This is significant where the lexical head has three arguments.

(15) [DNP$_{\text{subject}}$ [DNP$_{\text{object1}}$ [DNP$_{\text{object2}}$ [V X]]]]

With quantifier meanings given matching the required categories indicated in (12), then the DNP noun phrases in (15) will necessarily be interpreted with subject scoping over object1, and object1 scoping over object2. This is indeed the sole or natural scope found in many (and possibly all) languages. In (16), we have only one scope interpretation (allowing bound variable construal); in (17), both readings are available, but the one in (17b) is hard to process out of context.[18]

(16) a. Boris showed [each gibbon]$_k$ [its$_k$ new toy]

b. * Boris showed [itself]$_k$ [each gibbon]$_k$ (in the mirror)

(17) a. He sold [each slave]$_k$ [to herself]$_k$

b. He sold [herself]$_k$ [to each slave]$_k$

[18] We assume that the *to* in (17) is akin to a case-marker, and does not interrupt scope.

This result is not obtainable directly in standard CG except for SOV languages, because no displacement is admitted — Merge primarily constructs PF, so that the merge order (via directional slashes) would be as in (18), yielding the scope order 'subject > object2 > object1'.

(18) [DNP$_{subject}$ [[V DNP$_{object1}$] DNP$_{object2}$]]]

Within a minimalist analysis, the same erroneous c-command and scope expectations have forced additional assumptions and structure with both 'VP shells' and the assumption that a head can only have one internal argument currently assumed.

There are indeed languages (including Japanese and Mandarin Chinese) where the natural scope order is the only interpretation available for this merged order in the clause. In others, like English, the surface order gives the unmarked scope option, but alternative scope interpretations are permitted. Under the Functor First approach, deviations from natural scope must be due to differences in the order in which the DNP phrases are merged (e.g. by use of the composing combinator $\underline{\mathbf{B}}$), or to the content of what is merged (e.g. by type-lifting the verb). The proposals here, then, give a better match to the normal scope facts in simple clauses, and the options available, than either standard CCG or standard minimalism.

5.3 Particles and resultatives

For English, if the LF-merge order is SOVX, there must of course be some 'displacement' of the verb. There is prima facie evidence from English that the verb is merged at LF following the DNP arguments, but preceding any other complements. For example, the particle of an idiomatic verb-particle complex may occur in the position where we are claiming that the verb is merged. This is consistent with the verb-particle complex being drawn from the lexicon as such, and merged intact. Similarly, compositional resultatives will be merged as a unit, and may be displaced leftwards as a whole or with stranding.

(19) a. They **made** George **out** to be a criminal

b. They George [**make out**] [to be a criminal] (LF order)

(20) a. (i) He **hammered** the protruding nails **into the wood**

(ii) He **hammered into the wood** the protruding nails

b. He [the protruding nails] [**hammered into the wood**]

The data can of course alternatively be described by postulating a ‘VP shell’, but this seems to require stipulation. The data is directly explained by Functor First together with the leftwards displacement of some V-projection.

5.4 The position of clausal arguments

For internal clausal arguments, the prediction is that the ordering will be V or other lexical head followed by CP, even where noun phrases (DNP) precede the head. For many languages, such as Nupe and German, this is correct. It is languages like Japanese, where subjects and objects precede the verb, but CP does as well, as in (21), that now need an account.

(21) [$\text{DNP}_{\text{subject}}$ [$\text{DNP}_{\text{object}}$ [CP V]]]

There are three options available. If Japanese is Functor First, then for C, the TP operand must be displaced leftwards round C (and similarly for numerous other heads). If Japanese is Functor Final, then we expect DNPs to follow their second operand, so they should be clause-final. The correct order could be obtained by displacement of this operand rightwards for each DNP, but such large scale Pied Piping seems improbable. It seems more likely that the apparent deviance of Japanese is due to the nature of quantification in the language; but this is a matter for another paper.

Initial clausal subjects, as in (22), are also unexpected.

(22) [$_{\text{CP}}$ That John is late] [$_{\text{VP}}$ proves that he is not infallible]

The explanation is plausibly that they are obligatorily introduced by a null type-lifting determiner, like proper names.[19] A subject clause then must be a ‘DetCP’, just as argument noun-phrases are ‘DetNP’. We suppose that there is a null determiner comparable to ‘↑’ of section 4.2, related to finite T, as in (23):

(23) **determiner** category: T/(T/D)/C; type <t/<t/t>/t>
semantics: $\lambda x\ \lambda P\ Px$ for x of type <t> and P of type <t,t>

What forces the inclusion of a determiner? Here we need some equivalent of the ‘D feature of T’ (Chomsky 1995a: 232). Suppose that in languages like English which

[19] In Modern Greek, subject clauses are obligatorily introduced by a determiner.

must have subjects in Tensed clauses, T is lexical — a raising head. It has the category and simplified type shown in (24):[20]

(24) category: T / D / V type: <t /*u* /t> where *u* is the underspecified type[21]

Because the innermost selection of the Tense head is specified as D, it will not be possible to merge a bare clausal argument of category C in the subject position of the TP; but because the type is underspecified, it is possible to merge a subject which binds either a type <e> selection, as with normal noun phrase subjects, or, as here, a subject binding a type <t> selection. The categorial and type properties of the subject in a raising structure come from the selected VP. Then the relevant selection made by the VP must be [/D], not [/C], in order that the raising structure is possible. That this is indeed possible is confirmed by alternations such as that in (25), where a type <t> selection appearing in subject position is realised as a clause (detCP), as in (25a), or as a noun phrase (detNP), as in (25b):

(25) a. [That Nicholas had won the prize] was reported by the press

b. [The events] were reported by the press

With very few exceptions, a head that can select for a CP may also select instead for a DP. The few exceptions, such as *wonder* with *whether*, do not admit raising of the CP to subject position, as predicted:

(26) a. They wondered whether [Nicholas had won the prize]/*[the events]

b. *[Whether Nicholas had won the prize] was wondered

The explanation required for clausal subjects under our hypothesis makes predictions which are supported by independent data.

6 Conclusion

We have proposed a linearisation algorithm ‘Functor First/Last’ which operates uniformly on each instance of binary merge. The proposal makes essential use of the Categorial Grammar concept of a category, where the category includes

[20] There should of course be types giving a temporal dimension to the semantics.

[21] In previous work, we referred to the type here as allotting a ‘nil role’; what is intended is that the selection is syntactically and semantically real, but the Meaning Postulates associated with the head do not make reference to this selection. It follows that the type is not determined by the head.

information about the selections made by the head (or phrase). With these categories in place the whole of the apparatus of 'segments' of categories, relating to adjuncts, and the corresponding adaptations to c-command, can be dispensed with. Functor First/Last makes no reference to the notion of c-command. The algorithm is arguably both simpler and more consistent with the Minimalist Program than Chomsky's adaptation of Kayne's LCA proposal.

The linearisation we propose makes different predictions about underlying word order, so that work needs to be done on certain languages to substantiate its viability. However, it seems to us that the increased simplicity of the underlying assumptions is such that the proposal is worth pursuing.

References

Baker, Mark. 1988. *Incorporation: A theory of Grammatical Function Changing*. Chicago: University of Chicago Press.

Chomsky, Noam. 1986. *Barriers*. MIT Press.

Chomsky, Noam. 1995a. *The Minimalist Program*. MIT Press.

Chomsky, Noam. 1995b. Bare Phrase Structure. In *Government and Binding theory and the Minimalist Program*, ed. G. Webelhuth: 383-439. Blackwell Publishing.

Chomsky, Noam. 2000. Minimalist inquiries: the framework. In *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik*, ed. R. Martin, D. Michaels & J. Uriagereka: 89-155. MIT Press.

Chomsky, Noam. 2004. On phases. ms., MIT.

Chomsky, Noam. 2005. Three factors in language design. *Linguistic Inquiry* 36: 1-22.

Collins, Chris. 2002. Eliminating labels. In *Derivation and Explanation in the Minimalist Program*, ed. S. D. Epstein and D. T. Seely: 42-64. Blackwell Publishing.

Cormack, Annabel. 1999. Without Specifiers. In *Specifiers: Minimalist Approaches*, ed. D. Adger, S. Pintzuk, B. Plunkett, and G. Tsoulas: 46-68. Oxford University Press.

Cormack, Annabel, and Richard Breheny. 1994. Projections for Functional Categories. In *UCL Working Papers in Linguistics 6*: 35-61.

Cormack, Annabel and Neil Smith. 1999. Where is a sign merged?. *Glot International* 4: 20.

Cormack, Annabel and Neil Smith. 2000a. Head movement and negation in English. *Transactions of the Philological Society* 98: 49-85.

Cormack, Annabel, and Neil Smith. 2000b. Fronting: the syntax and pragmatics of 'Focus' and 'Topic'. In *UCL Working Papers in Linguistics 12*: 387-416.

Cormack, Annabel, and Neil Smith. 2005. What is coordination?. *Lingua* 115: 395-418.

Frampton, John and Sam Gutmann. 2002. 'Crash-Proof Syntax. In *Derivation and Explanation in the Minimalist Program*, ed. S. D. Epstein T. D. Seely: 90-105. Blackwell Publishing.

Hawkins, John A. 1994. *A Performance Theory of Order and Constituency*. Cambridge University Press.

Heim, Irene, and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Blackwell Publishing.

Kayne, Richard S. 1994. *The Antisymmetry of Syntax*. MIT Press.

Kayne, Richard S. 2005. Some notes on comparative syntax. In *The Oxford Handbook of Comparative Syntax*, ed. G. Cinque and R.S. Kayne: 3-69.

Kennedy, Christopher and Jason Merchant. 2000. Attributive comparative deletion. *Natural Language and Linguistic Theory* 18: 89-146.

Kihm, Alain. 2005. Noun class, gender, and the lexicon-syntax-morphology interfaces. In *The Oxford Handbook of Comparative Syntax*, ed. G. Cinque and R.S. Kayne: 459-512.

Larson, Richard K. 1988. Light Predicate Raising. In *Lexicon Project Working Papers 27*, MIT, Cambridge, MA.

May, Robert. 1985. *Logical Form: its Structure and Derivation*. MIT Press.

Sportiche, Dominique. 1994. Adjuncts and Adjunction. *GLOW Newsletter* 32: 54-55.

Sportiche, Dominique. 1995. Sketch of a reductionist approach to syntactic variation and dependencies. In *Evolution and Revolution in Linguistic theory*, ed. H. Campos, and P. Kempchinsky: 356-398. Georgetown University Press, Washington DC.

Starke, Michal. 2001. On the Inexistance of Specifiers and the Nature of Heads. http://ling.auf.net/lingBuzz/000004.

Steedman, Mark. 2000. *The Syntactic Process*. MIT Press.

Stroik, Thomas. 1990. Adverbs as V-sisters. *Linguistic Inquiry* 21: 654-661.

Wilder, Chris and Hans Gärtner. 1997. Introduction. In *The Role of Economy Principles in Linguistic Theory*, ed. C. Wilder, H. Gärtner, and M. Bierwisch: 1-35. Akademie Verlag Berlin.