Recursive linearization

Joost Kremers
University of Frankfurt
Institute for Cognitive Linguistics
Grüneburgplatz 1
60629 Frankfurt am Main, Germany
j.kremers@em.uni-frankfurt.de

Recursive Linearization

Joost Kremers
University of Frankfurt

Abstract

Linguistic structures are represented as hierarchical tree structures, although utterances are linear. To deal with this discrepancy, the grammar model needs to specify in which way linear order is derived from hierarchical structure. This article examines the idea that linearization is a PF procedure, implemented as a recursive procedure searching the hierarchical tree top-down for terminal elements. It is shown that we can describe linearization by positing one linearization principle and two parameters, one a modified head parameter, and one parameter to linearize adjuncts. The resulting system is applied to the domain of the Arabic and English noun phrases. Although these domains have vastly different linear orders, with the proposed model their syntactic structures turn out to be only minimally different.

As developed here, the model makes a fundamental prediction about language: order variation only exists in positions created by External Merge. That is, an element that is moved will invariably shift toward the left in the linear string. It is tentatively suggested that this property results from efficiency requirements of the parser, which would show that the language faculty is perhaps primarily but not solely adapted to the requirements imposed by the C-I interface.

1. Introduction[1]

Linearization is the procedure that derives a linear order from a hierarchical structure. In generative syntax, the grammatical structure of a phrase is represented as a tree structure that defines hierarchical relations. Originally, tree structures also defined precedence relations, but in more recent work starting with Chomsky (1995) this idea has been abandoned, and tree structures have come to be seen as hierarchical structures with no inherently specified precedence relations.

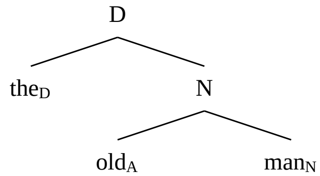
However, if tree structures do not define precedence relations, we need some mechanism that derives precedence relations from the hierarchical tree structure, because at certain levels of linguistic representation (specifically PF), precedence relations are relevant. Within a minimalist framework, two approaches toward linearization are in principle possible. We can assume that linearization is an inherent part of linguistic representation, which means that linearization is in some way defined in what is called *core* or *narrow syntax*. On the other hand, we can assume that linearization is a requirement imposed on language by the medium in which it is expressed. This would mean that linear order is not defined in core syntax; rather, it is derived from the hierarchical structure only where it is needed, i.e., at PF (or, in Chomsky's 2008 terms, the sensorimotor (SM) interface.)

Kayne (1994) represents an implementation of the first approach. In Kayne's model, linear order is derived from hierarchical structure by means of the *Linear Correspondence Axiom*, which maps the set of asymmetric c-command relations (defined on non-terminal elements in the tree) on the set of precedence relations (defined on the terminal nodes in the tree). Informally, if an element A asymmetrically c-commands an element B, all lexical material dominated by A precedes all lexical material dominated by B in the linear string. For Kayne, the LCA applies throughout the derivation, putting very specific restrictions on the structures that can exist during derivation.

The second approach, linearization at PF, finds its roots in Chomsky (1995), who adopts bare phrase structure as an alternative to X-bar theory. Chomsky argues that the only operations that the computational system needs are Merge and Agree. Merge is the operation that builds trees: it takes two elements and combines them into a larger structure. In Chomsky's view, the elements that Merge combines are

[1]The work presented in this article is based on earlier work that was part of my Ph.D. dissertation, and has benefited much from comments and suggestions of, and discussions with my supervisors, Kees Versteegh, Ad Foolen, and especially Eric Reuland, whose help and suggestions were invaluable. I would furthermore like to thank my former colleagues at the Graduate College "Sentence Types" at the University of Frankfurt, and two anonymous reviewer for their helpful comments and suggestions.

lexical items.[2] So for example, Merge can operate on *old*$_{\mathrm{A}}$ and *man*$_{\mathrm{N}}$ to form N(old, man), which can then be combined with *the*$_{\mathrm{D}}$ to form D(the, N(old, man)):

(1) 

Chomsky mentions that such trees do not have an inherent linear order. He states: "Nothing has been said about ordering of elements. There is no clear evidence that order plays a role at LF or in the computation from [the numeration] to LF. Let us assume that it does not." (Chomsky 1995: 334).[3] He then continues to argue that although the LCA crucially relies on X-bar levels, it is possible to reformulate the LCA in terms that make it compatible with bare phrase structure.[4] Here, Chomsky places the LCA at PF: "We take the LCA to be a principle of the phonological component that applies to the output of Morphology, (...)" (Chomsky 1995: 340).

In other words, Merge itself does not impose a linear order on the structures it creates, and none is defined in core syntax. A structure K(A,B) formed by Merge is identical to the structure K(B,A). At the same time, Chomsky maintains the general idea behind the LCA that if an element A asymmetrically c-commands an element B, A precedes B in the terminal string, although this assumption now only applies at PF.[5]

What I would like to do in this article is to take the idea that linearization is a PF phenomenon as a starting point and then look at what we would need in order to linearize tree structures. I will not take the LCA as a given, neither in core syntax as Kayne does, nor at PF as Chomsky or Moro (2000) suggest. Instead, I will only assume what can plausibly be assumed to be minimally required, starting with a bare phrase structure framework as outlined in Chomsky (1995) and later work.

As we will see, one of the advantages that the approach developed in this article

[2]Where it should be noted that functional heads are considered lexical items as well.

[3]Strictly speaking, of course, assuming that order does not play a role at LF does not necessarily imply that there *is* no order at LF, but this does seem to be the most parsimonious assumption. Cf. also Chomsky (2008), where he argues that Merge forms a set {X,Y}, not an ordered pair ⟨X,Y⟩, and "that order does not enter into the generation of the C-I interface, and that syntactic determinants of order fall within the phonological component." (Chomsky 2008: 5).

[4]Obviously, as Chomsky also notes, the theoretical motivation for the LCA in that it accounts for several essential properties of X-bar theory, falls away in this reformulation, and only the empirical motivations remain.

[5]A far more elaborate proposal to incorporate the LCA into PF is made by Moro (2000).

has compared to the antisymmetric approach is that it does not rely on large-scale movement operations and undesignated functional heads in order to account for specific word order effects. This becomes particularly conspicuous when we examine so-called *roll-up* movements, required in antisymmetric analyses to account for certain adjective orderings. The PF-approach discussed here handles roll-up movements elegantly, without requiring additional movement operations.

This article is structured as follows: in Section 2 I introduce the linearization procedure that I envisage. Section 3 shows how a straightforward implementation of this procedure can account for word order differences between the English and the Arabic noun phrase. In Section 4, I discuss how the linearization procedure from Section 2 can account for some well-known adjective ordering universals. Finally, Section 5 presents a comparison of the procedure developed here with Kayne's LCA, discussing in more detail the essential differences between the two approaches.

2. Linearization as a PF procedure

2.1. Recursive Linearization

The central question to be answered in this section is the following: how can we linearize a tree structure that represents a linguistic expression? This question can be divided into two subquestions: a) what are the elements in the tree that we must spell out; and b) how do we find those elements?

When Merge creates a structure, the resulting structure is compound. The elements from which this compound structure is formed can be compound themselves, but they can also be simplex, i.e. not formed by Merge.[6] Such simplex elements are the terminal elements of the tree, and it is these elements that we are looking for: they are the ones that must be spelled out.

The second question, how to search for these elements, is easily answered: linguistic trees are binary trees, for which various search methods exist (see, e.g., Winston 1992; Bratko 2001). Of these methods, the so-called *depth-first* search is the only one compatible with standard syntactic assumptions.[7] Depth-first search is a recursive search method, in the sense that it starts out at the root node of the tree and then works its way down by applying itself to each of the two subnodes.

Let us make the search procedure more explicit. A recursive linearization procedure ("RLin" for short) will consist of the following steps:

[6] They can be compound elements on another level, that is, formed by some operation other than Merge (e.g. in the lexicon). For Merge, however, they are simplex.

[7] See Kremers (2003) for some discussion.

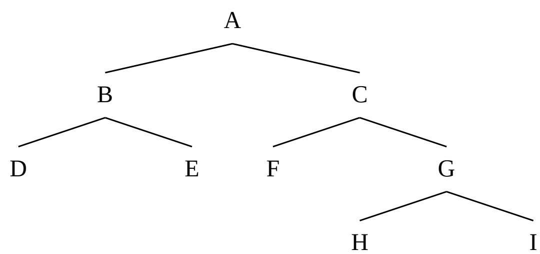
(2) linearize tree T:

a. if T is a terminal element, spell out T.[8]
b. otherwise, take subnodes A and B of T.
c. linearize either A then B, or B then A.

Step c is to be understood as follows: if the order A-B (B-A) is chosen, subnode A (B) has to be spelled out *in its entirety* by applying the procedure in (2) to it, before RLin can start with subnode B (A). Furthermore, only when subnode B (A) has been completed *in its entirety* as well, does (this application of) RLin terminate. This is where the procedure is recursive: it applies itself to each of the subnodes in order.

To see how this all works, take the tree in (3):

(3)



In (3), the terminal elements are D, E, F, H and I. When RLin is at the top node A, it must either linearize B first and C second, or C first and B second. If B is taken first, the procedure applies itself to the node B(D,E), at which point RLin must again take one of the subnodes D or E to linearize first. Whichever is taken, RLin will encounter a terminal element, which can be spelled out. When both D and E have been spelled out (in whichever order), the node B has been linearized completely. This completes the first subnode of node A. RLin will now start with the second subnode of A. This node C is dealt with in the same manner, which will eventually lead to the spelling out of all the terminal elements in it, i.e., F, H and I.

Note that the decision whether to linearize B before C or C before B has a direct influence on the order in which the elements are found: if B is searched first, D and E will be found and spelled out before F, H and I. If C is searched first, F, H and I will be spelled out before D and E. This is an important principle of RLin: a node is always searched in its entirety and consequently all the terminal elements in it are spelled out before RLin proceeds with its sister.

[8] Note that I use the term “spell-out” with the specific sense of putting a terminal element in the linear string, which differs from the standard meaning of the term. With traditional “spell-out” now being replaced by the notion TRANSFER, the risk of confusion should be small.

2.2. The order between subnodes

The procedure described in the previous section would seem to be the essential part of *any* PF procedure to linearize tree structures. However, at this point, we do not yet have enough to actually linearize a tree. Note that in (2c) a choice must be made: a node K(A,B) must be linearized either as A-B or as B-A. This choice is of course essential, without it there is no linearization. Furthermore, it crucially determines the order in which the terminal elements end up in the linear string, that is, it crucially determines word order.

So let us see how the procedure can make this choice. Ideally, RLin should be able to make this decision locally, that is, without using any information that is not available at the node that it is processing. Let us say that all the information that is available during the syntactic computation is also available when RLin starts processing the tree.[9] This means that the information in (4) is all available to RLin when it processes a node K(A,B):

(4) a. the category of K.
 b. which of the two subnodes projects, A or B.
 c. whether the projecting node is a terminal element (i.e. a head).

Obviously, the label of a node K must contain enough information to determine its category, so (4a) is known. In a bare phrase structure approach, the label of K actually indicates which of its two subnodes projects, yielding (4b). A node that is a terminal element is a node that is simplex, i.e., not formed by Merge.[10] This gives (4c).

With this information, we can formulate the following linearization principle:

(5) Principle P': Linearize the non-projecting subnode first.[11]

Under P', a structure of the form [Spec [H Comp]] will be linearized as Spec-Comp-Head: in the node [Spec H′], the specifier is the non-projecting subnode, so that it is linearized first. In the node [Head Comp], the complement is obviously the non-projecting node, so that a Comp-Head order results.

[9] Any procedure that would strip off information that is not necessary at PF is in fact a part of PF, because it is performing an operation that is essentially relevant to PF. So we can simply say that RLin is the first PF operation to apply.

[10] Such a node will also have a lexical (morphological and phonological) representation, because it must have come from the lexicon or be composed through certain morphological operations. There is, however, good reason to believe that this lexical representation is not (directly) present in syntax, but is obtained at some point during PF derivation. See e.g. Halle and Marantz (1993) and Ackema and Neeleman (2004) for discussion.

[11] P' is to be read as P-bar, i.e., “non-projecting node”.

For specifiers, P' is all that we need, as they are generally realized before their heads.[12] For head-complement structures, P' is not adequate in itself, however: while complement-head structures are common, head-complement structures are at least as common, and probably more so. We therefore need a way to have RLin linearize a head (i.e. terminal element) immediately when it sees it. That is, we need a head parameter:

(6) Head parameter H:

H^+ : Linearize a head immediately.
H^- : Do not linearize a head immediately.

The setting H^+ for the head parameter overrides the effect of P'. Instead of linearizing the non-projecting node, RLin will deal with the head first, and only then will it linearize the non-projecting node, which results in Head-Comp orders. The setting H^- obviously has no such overriding effect, yielding Comp-Head orders.

We can even go one step further and say that the setting H^+ is the default. That is, a child will assume that every category has the setting H^+, unless it finds clear evidence in the primary linguistic data for the opposite order. Only if the primary linguistic data shows that a specific head consistently follows its complement will the language-learning child switch the setting to H^-. This may account for the observation that newly created languages, e.g. pidgins and creoles, have a strong preference for VO orders (Holm 2000). Because a child that forms a creole out of a pidgin does not hear any consistent evidence for OV structures, it will use the default setting, which yields a VO structure. At the same time, we can understand how it is possible for OV languages to exist and to be stable over long periods of time: because OV structures occur abundantly, a child acquiring the language has sufficient evidence to switch the order from its default.

We know that within one language, the head parameter H can have different values for different categories. For example, in Dutch and German, V follows it complement, whereas C and D precede them. This means that the value of H can be specified independently for different categories. This is potentially problematic, because syntactic studies show that the number of (functional) categories in language can be quite extensive. A clause does not only have C, T, *v* and V, there is also evidence for projections like Asp, Neg, Top, Focus and perhaps others. In the noun phrase, apart from D and N there may also be categories such as K, Poss, Num and Gen.

A simple calculation reveals that if we have six categories in the noun phrase (K D Poss Num Gen N), and each category can specify its own value for H, we obtain

[12] Unlike Kayne, I do not treat adjuncts as specifiers. See Section 2.3 for discussion.

$2^6 = 64$ different possible word orders. However, in any one language, many of these orders will not be distinguishable for the simple reason that most languages do not have overt, independent markers for all of these heads. Therefore, we can limit the group of categories that can specify their own value for H in a natural and straightforward way by saying that only those categories that have an *independent* morphological form have this ability, whereas a category that is not phonologically marked does not. Instead, such a category must adopt the default setting H^+.

One advantage of limiting the set of heads that can specify their own value for H to those heads with independent morphological form is that it should be relatively easy for a child acquiring its mother tongue to determine for which heads it must establish the ordering, and furthermore, what the ordering must be. Once a child has determined that for example X is a head and YP its complement, it only needs to look at the linear order in which X and YP occur to determine the value of the head parameter for X.[13]

2.3. Adjuncts

The language system as a whole must be able to distinguish between selected and non-selected elements. The term "selected" refers to elements that are merged to satisfy some feature, either a selectional feature (a verb selecting its arguments), an Agree feature, or an *edge feature* (raising elements into the C-domain), and comprises complements and specifiers. "Non-selected" refers to those elements that are not merged to satisfy some feature, and comprises adjuncts.

The language system must make this distinction, if only because it is relevant for semantics: if Merge is interpreted semantically as functional application, selected elements are arguments, while the selecting element (the head of the projection) is the functor. In adjunction structures, the reverse is true: the adjunct is the functor, while the element being adjoined to (the element that determines the category of the newly created structure) is the argument. Furthermore, adjuncts are not assigned theta roles by a verb/noun, and they show specific reconstruction effects that arguments do not share (cf. e.g. Lebeaux 1991).

However, the relevant question for the present discussion is not so much whether the selected/non-selected distinction is relevant for the language system as a whole, but rather whether it is relevant for linearization. Kayne (1994) argues that it is not. He claims that adjuncts are essentially specifiers of designated projections, so that they are selected, in the sense intended here. However, Kayne's reasons for doing so are theoretical: because an adjunction structure creates a new segment but not a new category, no asymmetric c-command is established between the adjunct and

[13] Obviously, the linear order of X and YP must occur with a certain consistency. A single occurrence of OV does not guarantee that the language is indeed OV.

the structure adjoined to, so that the entire structure does not comply with the LCA. Treating adjuncts as specifiers solves this problem.

Because the primary motivation for treating adjuncts as specifiers is theory-internal to the antisymmetric approach, there is no *a priori* reason to do the same in a PF-linearization approach. Therefore, we may ask ourselves whether there is empirical evidence to treat adjuncts as distinct from specifiers. In actual fact, there is: while specifiers are almost invariably placed to the left of their heads, it is possible for adjuncts to consistently appear to the right of their heads, as shown by postnominal adjectives and postverbal adverbs.

Antisymmetric approaches deal with this empirical evidence by postulating various movement operations. Obviously, movement operations are not incompatible with RLin, but RLin does offer us a different way to account for such word order phenomena. I will therefore take the ordering variation observed with adjuncts as an indication that adjuncts are treated differently than specifiers by the linearization procedure: adjuncts are not subject to principle P'.

There are actually theory-internal grounds for making this assumption as well: in an adjunction structure such as [$_{\mathrm{XP}}$ Adj XP] the phrase XP does not actually project. That is after all what it means for a structure to be an *adjunction* structure: it creates another segment of XP, but not a projection of it. A structure that does not project cannot be subject to a principle such as P', that explicitly refers to projection.

In order for RLin to distinguish between specifiers and adjuncts, the distinction must be made in syntax, i.e., adjuncts must have a distinct syntactic structure.[14] Chomsky (2004) proposes to distinguish between set-Merge ("normal" Merge), which creates a structure $\{\alpha, \{\alpha, \beta\}\}$, and pair-Merge (adjunction), which creates a structure $\{\langle\alpha, \alpha\rangle, \{\alpha, \beta\}\}$. This provides sufficient means for a linearization procedure to distinguish adjuncts.[15]

The most straightforward way to deal with the considerable variation in placement of adjuncts is to say that their linearization is parametrized. That is, if RLin encounters an adjunction structure K(A,B), the order in which A and B are linearized is determined by a parameter, which we can call the *adjunct parameter*.

The considerations concerning the learnability of the head parameter also apply to the adjunct parameter. We must limit the ability to specify a value for the adjunct

[14]The same argument can be made on the basis of the semantic distinctions discussed above, obviously.

[15]Note that whether adjunction takes place "early" or "late" (cf. Lebeaux 1991; Stepanov 2001 and much related literature) is not relevant for the question at hand. Whether early or late, adjunction must take place before TRANSFER: if it did not, the semantic and phonological systems might end up with different tree structures. Since linearization takes place after TRANSFER, the adjunction point has no bearing on linearization.

parameter to independent morphological forms. There is a small difference, however. Unlike the head parameter, the adjunct parameter presumably does not have a default form. Because of this, we may say that a head without an independent morphological form is forced to adopt the setting of its complement.[16]

Summarizing, we see that RLin only needs one principle, P', and two parameters to account for the ordering variability in language: a head parameter and an adjunct parameter. All the other information that RLin needs in order to linearize a tree structure is available in that tree structure itself.

Having established how the procedure operates, we must now ask the question on which trees it operates. RLin obviously operates on the trees it is given, and in a minimalist model, this means the structures that are sent to the PF interface, (*transferred*, as Chomsky (2008) puts it), i.e., on phases. A phase that has been linearized and spelled out presumably yields some phonological structure that can be integrated in a higher phase. For RLin, it would suffice to say that the resulting phonological structure is in the tree and is treated as a terminal element. That is, in the higher phase, it will at some point be found by RLin and placed in the linear string, just as all other terminal elements, as described above.[17]

3. Implementation

In order to see what the theory sketched above gives us, we must try and apply it to linguistic data. In this section, I examine the domain of the English and Arabic noun phrases, in order to determine how a procedure such as RLin can help us account for the word order of these domains, and the differences in word order between the two languages. The differences in word order in the DP between the two languages are large: English has possessor-noun and adjective-noun, Arabic has noun-possessor and noun-adjective. Furthermore, Arabic adjectives show a so-called *mirror image* order, which antisymmetric approached deal with through *roll-up* movement, so that an LCA-based account for English and Arabic adjective orderings proposes vastly different structures for the two languages. As we will see, however, RLin allows us to develop structural accounts for the two domains that differ only minimally from each other, and that the one difference that we must

[16] As we will see in Section 4, this assumption is not as *ad hoc* as it may now seem.

[17] Note that this view may open up the way to a PF-treatment of relative clause extraposition and heavy-NP shift. Assuming that the extraposed/shifted element is a phase, it would appear in its containing phase as a terminal element, as described. We may then be able to state that RLin has an option to *postpone* spell-out (i.e., inclusion into the linear string) of a terminal element that is "heavy", in some phonologically defined sense, until all other material in the containing phase has been spelled out.

assume is well motivated.

3.1. *The structure of the Arabic and English noun phrase*

Let us begin by examining the word order differences between noun phrases in the Arabic and English. The structures that I focus on in this article are the ones in (7):

(7) a. the house
b. the man's house
c. the red house
d. the man's red house
e. the beautiful red house

(7) contains structures with just a noun (7a), a possessed noun (7b), a noun plus an adjective (7c), a possessed noun plus an adjective (7d), and a noun with two ordered adjectives (7e). Now consider the Arabic equivalents of these phrases:

(8) a. *al-bayt-u*
the-house-NOM
b. *bayt-u al-rağul-i*
house-NOM the-man-GEN
c. *al-bayt-u al-'aḥmar-u*
the-house-NOM the-red-NOM
d. *bayt-u al-rağul-i al-'aḥmar-u*
house-NOM the-man-GEN the-red-NOM
e. *al-bayt-u al-'aḥmar-u al-ğamīl-u*
the-house-NOM the-red-NOM the-beautiful-NOM

When we look at the Arabic examples, we notice a few things. First of all, Arabic has the peculiarity that a possessed noun, such as *bayt* 'house' in (8b), does not have a definite determiner.[18] Furthermore, the word order of the various phrases is quite different from their English counterparts. In Arabic, the possessor follows after the possessed noun, whereas in English it precedes it. Adjectives also follow the noun, while they precede the noun in English. Note also that Arabic adjectives are not only post-nominal, their order is also a mirror image of the English order: English has *beautiful red house* in (7e), while Arabic has the equivalent of *house red beautiful* in (8e).

And more intriguingly, when a noun is possessed and also modified by an adjective, the adjective follows *after* the possessor. In other words, the adjective is

[18]Note that it also lacks an indefinite marker, which most (unpossessed) nouns take in the form of a suffix *-n*.

not allowed to intervene between the possessed noun and its possessor. This contrasts with English, in which an adjective can appear between the possessed and the possessor noun, as happens in (7d).[19]

I follow Abney (1987) in assuming that the noun phrase is headed by a D projection, and that the nominal head N is in the complement of this D. These are not the only heads in the noun phrase, however. When we examine the evidence, it turns out that there must be others.

First of all, Arabic and English both have a very particular possessive structure. The term "possessive" is somewhat misleading, in that it implies a specific semantic relation. Here, the term is to be understood as a purely syntactic notion, referring to a specific syntactic construction, without any implications for the semantic interpretation of this construction.

The possessive construction in Arabic is the so-called *construct state* (see, e.g., Ritter 1991; Borer 1999; Fassi Fehri 1993; Kremers 2003):

(9) *bayt-u al-rağul-i*
house-NOM the-man-GEN
'the house of the man/the man's house'

Arabic nouns normally have either a definite determiner *al-* or an indefiniteness suffix *-n*. The possessed noun in construct state lacks both, as shown in (9): *bayt* only has a case ending, no (in)definiteness marking.

English has the well-known prenominal genitive, usually called *Saxon genitive* (Abney 1987; Zribi-Hertz 1997):

(10) the man's house

At first sight, these two possessive structures are very different. However, on closer examination, it turns out that they have very similar properties, and RLin allows us to describe both constructions with one single structure.[20] The first property that both constructions share is the fact that the possessed noun does not have a determiner (either definite or indefinite) on its own. For English, we can demonstrate this with the following example:

(11) (*the) that man's (*the) houses

[19] An adjective that intervenes between the possessed and the possessor in English of course modifies the possessed noun, because English adjectives precede the noun. The important thing to note is that something can intervene between the possessor and the possessed in English, but not in Arabic.

[20] The present discussion is not meant as an exhaustive treatment of both possessive constructions, but rather as an illustration of the linearization model and how it can help create a unified analysis across languages. A more thorough analysis may likely reveal more structural differences between English and Arabic, but I do believe the general idea behind the analysis here can be maintained.

In (11), we see that the possessed noun, *houses*, cannot take a determiner: in both positions indicated, *the* is ungrammatical. Note that the one determiner that does occur in the phrase, *that*, belongs to the possessor noun *man*: *that* is a singular form, like *man* and unlike the plural noun *houses*.

For Arabic, the point is even easier to make. As just mentioned, a non-possessed noun in Arabic is always marked for definiteness: a definite noun takes the determiner *al-*, an indefinite noun takes the suffix *-n*:

(12) *bayt-u-n;* *al-bayt-u*
house-NOM-INDEF; the-house-NOM
'a house; the house'

A possessed noun, however, does not take either marker:

(13) a. *bayt-u* *al-rağul-i*
house-NOM the-man-GEN
'the man's house'

b. *(*al)-bayt-u-(*n)* *al-rağul-i*
(*the)-house-NOM-(*INDEF) the-man-GEN

(13) shows that the possessed noun *bayt* 'house' cannot take *al-* nor the suffix *-n*. We see, then, that Arabic and English share this property: the possessed noun in a construct state and in a Saxon genitive cannot take a determiner.

A second property that the two languages share is the fact that although the possessed noun does not have a marker for definiteness, it does have a definiteness feature. It "inherits" this feature from the possessor noun. In English, this can be demonstrated with the following two examples (from Grimshaw 1990):

(14) a. *There is the man's shirt on the chair.
b. There is a man's shirt on the chair.

The clause in (14) is an existential structure, which has the requirement that the subject must be indefinite. In (14), the subject is a possessed noun *man's shirt.* As (14a) shows, when the possessor noun is definite, the sentence is out, but when the possessor noun is indefinite, as in (14b), the structure is grammatical. We can only account for this if we assume that the definiteness of the possessed noun, which is after all the head noun of the noun phrase, is inherited from the possessor noun. (We will see below how we can account for this inheritance.)

In Arabic, the phenomenon is even easier to demonstrate. First, note that attributive adjectives in Arabic agree with the noun they modify not only in gender, number and case, but also in definiteness:

(15) a. *rağul-u-n* *ṭawīl-u-n*
man-NOM-INDEF tall-NOM-INDEF
'a tall man'

b. *al-rağul-u* *al-ṭawīl-u*
the-man-NOM the-tall-NOM
'the tall man'

Adjectives have the same morphology as nouns. That is, they take the definite determiner *al-*, or the indefiniteness suffix *-n*. As (15) shows, the definiteness marker that the adjective takes is the same as the definiteness marker of the noun. Given this mechanism, we can see what the definiteness feature of a possessed noun is by modifying it with an adjective and seeing what definiteness marker this adjective takes.

(16) a. *bayt-u* *ṭalabat-i-n* *kabīr-u-n* / **al-kabīr-u*
house-NOM students-GEN-INDEF large-NOM-INDEF / *the-large-NOM
'a large student house'

b. *bayt-u* *al-ṭalabat-i* **kabīr-u-n* / *al-kabīr-u*
house-NOM the-students-GEN *large-NOM-INDEF / the-large-NOM
'the large student house'

In (16a), the possessor noun *ṭalaba* 'students' is indefinite, and the adjective *kabīr* 'large' must copy this indefiniteness feature. Note that in other aspects, *kabīr* agrees with *bayt* 'house': they are both singular and nominative, while the possessor noun *ṭalaba* 'students' is plural and genitive. Furthermore, semantically the adjective modifies the head noun, not the possessor. In other words, we must conclude that the adjective agrees with the head noun in definiteness as well, and that therefore the head noun must inherit the definiteness feature of the possessed noun.[21]

There is a straightforward way to account for both the lack of definiteness marker on the possessed noun and the definiteness inheritance. Given that definiteness is obviously linked to possession in these constructions, it is a common assumption

[21] Intuitively, it may appear more accurate to say that it is not the definiteness feature of the head noun but rather the definiteness feature of the entire construct state that is copied onto the adjective. There is no contradiction here, however: we will see below that syntactically, we must describe the relevant definiteness feature as a feature of the head noun, but at the same time we can ascribe the interpretation of that feature to the entire construct state, especially in cases such as (16) above, where the construct state has a compound-like meaning 'student house' rather than 'the house of the students' (although it *can* in principle mean both.)

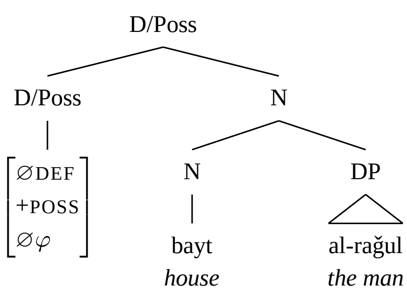
(going back to at least Abney 1987) that there is a special D-head involved, which I analyze as a head with combined definiteness and possessive features, D/Poss for short.[22]

The idea of a "combined" head stems from Giorgi and Pianesi (1997), who argue that it is possible for two features to project onto a single head, and furthermore that when this happens it is also possible that the value of one of the features is fixed. Following this assumption, we can say that the DEF feature and the POSS feature project onto a single combined D/Poss head in the Arabic construct state and the English Saxon genitive, and that in this case the DEF feature is unvalued at the onset of the derivation.[23]

I furthermore assume that the possessor noun of a possessive construction is in the complement position of N. This follows from the assumption that the external argument of a noun is the referential argument *R* (Higginbotham 1985; Zwarts 1992). Although possessors are often taken to be external arguments, it should be noted that in deverbal nouns, internal arguments of the underlying verb can be possessors as well, without requiring any sort of passive morphology (cf. Longobardi 2001 for some discussion). This is easily accounted for if we assume that all possessors are in fact internal arguments.[24]

With these assumptions, the structure of the Arabic tree is the one presented in (17):

(17)



In the tree in (17), the D/Poss head has unvalued features: DEF and φ. There

[22]The main point of this article does not hinge on the details of the DP-analysis, and therefore a detailed motivation for the structure adopted will not be provided here. The interested reader is referred to Kremers (2003).

[23]On the assumption made by Chomsky (1995) that functional heads are lexical elements, we can compare this kind of D/Poss head to a lexical item such as *men*, in which a set of semantic features is combined with a number feature, and where the number feature is fixed to plural. See also Reuland (1990) for the idea that two heads can combine to form a single head. Reuland makes the argument for V and I in Dutch.

[24]Again, the main point of this article does not hinge on this assumption, and it is made here primarily for consistency with the DP-analysis being assumed.

is very little evidence in Arabic that D/Poss has a set of φ-features, but languages such as Turkish and Hungarian show it clearly (cf. Abney 1987; Szabolcsi 1994). I assume that the φ-features on Poss in Arabic are covert, just like the φ-features on T in many languages.[25]

The set of φ-features and the DEF feature on D/Poss are unvalued when the head is merged. As such, the head is active and will probe its complement for a match. It finds one in the possessor DP, which has a set of valued φ-features, and an unvalued CASE feature, making it active. An Agree relation is established between D/Poss and the complement DP to value the φ-features of D/Poss. But note that D/Poss also has an unvalued DEF feature, and that the complement DP has a valued DEF feature. As Chomsky (1995) argues, when an Agree relation is established, all the unvalued features of both the probe and the goal are valued when they can. Therefore, the Agree relation that is established between D/Poss and the possessor noun also values the DEF feature on D/Poss. Through this mechanism, the apparent "inheritance" of the DEF feature is just another instance of the familiar Agree operation.

The structure of the Saxon genitive in English is basically the same as the Arabic construct state. There is, however, one difference, which is revealed when we look at the linearization of these structures. Before we discuss the linearization of English, however, we will first turn to Arabic.

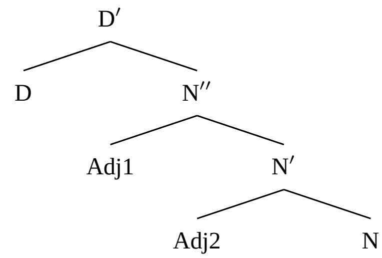
3.2. *The linearization of Arabic*

The hierarchical structure of the non-possessed noun is rather straightforward. I assume that adjectives are adjoined to some projection inside the noun phrase, although I remain agnostic on the exact projection. For ease of exposition, I will simply assume that all adjectives are adjoined to NP.[26]

[25]Alternatively, one could argue that there are no φ-features on D/Poss in Arabic (or English, for that matter). The Agree relation would still be established on the basis of the unvalued DEF feature. Note, by the way, that there is one small fact that may be taken as evidence for φ-features on D/Poss in Arabic: with some quantifiers that head a construct state, the head of the construction takes feminine form when the quantified noun is masculine and vice versa.

[26]Although bar levels have no theoretical meaning in bare phrase structure, I indicate them in the trees below to make it easier to refer to the different nodes.

(18)



The word order that we observe for (18) is (19):

(19) D N Adj2 Adj1

What would be the parameter settings that would derive the order in (19)? Note first of all that the D head precedes its complement. This establishes H^+ for D. Since N does not have a complement, we cannot determine the setting for H for it at this point.

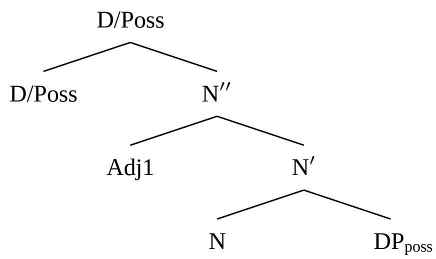
Because the only adjuncts in the phrase are adjoined to N, the value of the adjunct parameter can only be established for N. It must be set to adjunct-second. To see why this is so, consider the linearization procedure working its way down the tree: it starts at the root node D′ which, as we have just established, has H^+. As a result, in the root node D is linearized immediately, before N″, yielding a structure in which D is initial. In N″, the adjunct-second setting requires that N′ is linearized before Adj1, and in fact, the same thing happens in N′: because Adj2 is an adjunct, the adjunct parameter forces it to be linearized after N.

So at this point, N is spelled out. With that, RLin has reached the lowest node of the tree, but it has not finished linearizing it. Only one subnode of N′ has been completed. At this point, then, the second subnode of N′, which is Adj2, is linearized, giving us D N Adj2. With that, the entire node N′ has been linearized, meaning that the first subnode of N″ is now completed, and that the second subnode, Adj1, can now be linearized. When it has, N″ is completed, and since N″ was the second subnode of the root D to be linearized, the entire tree has now been processed. The end result is D N Adj2 Adj1, which is exactly what we find in (19).

Note that the adjunct-second setting is essential for obtaining this result. If the adjunct parameter were set to adjunct-first, RLin would linearize the two adjectives first in their nodes, rather than second, and that would put them not only in a different position in the linear string, but also in a different order with respect to each other. In N″, RLin would first linearize Adj1 before continuing with N′, and similarly in N′, RLin would linearize Adj2 before N. In other words, an adjunct-first setting would yield the order D Adj1 Adj2 N. As we will see below, this is how the adjective order in English is derived.

If we now turn to construct state nouns, we find that its word order properties fall out automatically, even the peculiar fact that the possessor noun separates the head noun from its modifying adjective. The structure of the possessed noun is (20), as we have just seen:

(20)



The corresponding word order is (21):

(21) D/Poss N DP$_{\text{poss}}$ Adj1

Because D is H^+, the D/Poss head will be linearized first. Note, however, that it is actually phonologically empty, as seen in the structure in (17) and that we therefore do no see it in the linear structure.[27] After D/Poss, RLin will continue with N″. Here, because of the adjunct-second setting, N′ will be linearized first. In N′, N must be linearized first and DP$_{\text{poss}}$ second. This means that we must assume that the head parameter for N is set to H^+ as well.

RLin has now completed N′, and with that the first subnode of N″. Therefore, it can now proceed to linearize the second subnode of N″, the adjunct Adj1. As a result, Adj1 follows all the material in the NP, which includes the genitive complement of the noun.

We see then that one single parameter setting, the adjunct-second setting of the adjunct parameter, is responsible for three salient facts in the Arabic noun phrase: the post-nominal position of adjectives, the mirror-image order of multiple adjectives, and the peculiar position of the adjective modifying the head noun in construct states.

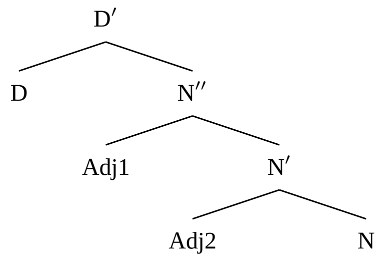
3.3. The linearization of English

Let us now turn to English. As I indicated, the initial assumption is that the tree structure for the English noun phrase is identical to the one for Arabic. Yet, the

[27] In fact, in Kremers (2003) I argue based on word-order phenomena in deverbal nouns that in construct state constructions, N moves to D/Poss. In structures of the type in (21), both heads appear adjacent in the linear structure, so this movement does not result in any visible word order change, and is therefore ignored here.

word order is different. First, consider the structure of the non-possessed noun. Its tree structure and linear order are the following:

(22) a.



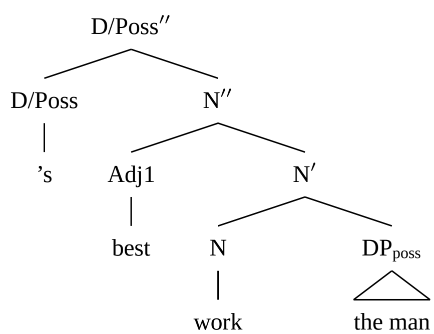
b. D Adj1 Adj2 N

Like Arabic, the English definite determiner precedes its complement. Therefore, D has the setting H^+. Unlike Arabic, the adjectives appear before N, so we assume an adjunct-first setting for N. With these settings, we can easily see that we derive the correct order from the tree in (22a). RLin starts at the top, with the DP. Because the category D is H^+, the head D is linearized first. Next, N″ is linearized. Here, because of the adjunct-first setting, Adj1 is linearized before N′. In N′, the same thing happens: Adj2 is linearized first. Only then will N be linearized. The resulting order is the one shown in (22b).

This is an important result. We see that we can actually derive the order of the English and the Arabic noun phrases from one and the same hierarchical structure. The only thing that is different between the two languages is the setting for the adjunct parameter.

Let us now look at the structure of the possessed noun in English. Again we will start out with the assumption that the tree structure is the same as in Arabic. We do note one important difference, however: in Arabic, the head D/Poss is not overt, but in English it is: the element -*'s* appears in this position:[28]

(23) a.

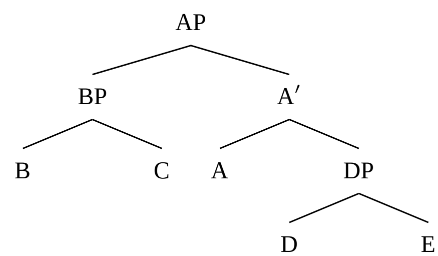


[28]Abney (1987) suggests that -*'s* may be a functional projection. See also Kayne (1993) and Zribi-Hertz (1997) for a similar position.

b. DP_{poss} D/Poss Adj1 N-Num
c. the man's best work

The reader will notice that there is a problem with this linear order, regardless of the settings of the linearization parameters. In the hierarchical structure, N and DP_{poss} are sisters, but in the linear structure, they are not adjacent: Adj1 and D/Poss (the element *-'s*) intervene. To see why this is a problem, consider the following tree:

(24)



If BP is a selected specifier, (24) will yield as surface order either BP-A-DP or BP-DP-A, depending on the head parameter for the category A. If BP is an adjunct, two more orderings become possible: A-DP-BP and DP-A-BP.

Note that in all of these orders A and its complement DP are adjacent. This follows from the fact that RLin searches a tree branch by branch: RLin will always linearize A and DP one after the other, either in the order A-DP or in the order DP-A, because they are both subnodes of the same node, A′, and because RLin always finishes a node completely before continuing with its sister node.

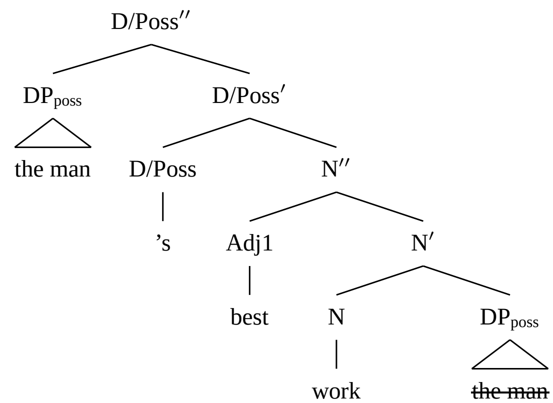
This is in fact an important observation: because of the way RLin operates, two sister nodes will *always* end up adjacent in the linear string. Elements that are not in either of the two sister nodes cannot intervene. Yet, this seems exactly what is happening in (23): N and its complement DP_{poss} are *not* adjacent in the linear structure, even though they are sisters in the tree structure.

We therefore conclude that either N or DP_{poss} is not in its base position in English. The most obvious choice is of course to say that DP_{poss} has moved to Spec,D/Poss, because, as we have seen, there is an agreement relation between D/Poss and the complement DP. We know that this agreement process exists because we can observe one result of it: definiteness inheritance. We simply conclude that in English, this Agree is followed by Merge.

This means that there are two differences between Arabic and English: in Arabic, N has the setting adjunct-second, while in English it has adjunct-first, and furthermore, the Agree relation between D/Poss and the complement DP is followed by Merge in English, but not in Arabic.[29]

Let us see what the structure of the possessed noun looks like with this movement, and how linearization takes place:

(25)



In order to linearize this tree, RLin starts at the top. Because DP_{poss} is a specifier, not an adjunct, (it has been moved after an Agree operation) principle P' applies, which means it is linearized first. We obtain the string *the man...* Then, D/Poss′ is linearized. This node consists of the head D/Poss and its complement. Because D/Poss has the setting H^+, the head is linearized first, and we obtain the string *the man's...* Then RLin moves to N″, which contains an adjunct, Adj1, and N′. Because N has the setting adjunct-first, Adj1 is linearized, and we obtain *the man's best...* Then, N′ is linearized. It contains the NP, but since the possessor DP has moved out, N is the only morphologically overt material that is left. After it is linearized, we have obtained the string *the man's best work*, which is exactly what we need.

[29] The reason why this movement takes place in English is not clear. Ideally, every movement operation has a semantic motivation, which does not seem to be present here. Note, however, that there is at least one other movement that has to date not received a satisfying semantic motivation: the movement of the subject to Spec,TP. The movement of the possessor noun to Spec,D/PossP is similar enough to this that we can attribute it to a nominal EPP feature. Alternatively, we may be able to argue that the movement takes place because the D/Poss head -*'s* needs to cliticise onto another element. However, this would place the movement at PF, which may not be desirable. I will leave the matter to future research.

4. RLin in the language system

4.1. Adjective ordering universals

In the examples of the previous section, there is one in which two adjectives modify one noun: *a beautiful red house*. In such cases, there is often a preferred order for the two adjectives: *a beautiful red house* is fine, but #*a red beautiful house* sounds rather bad. (See e.g. Laenzlinger 2000 and Sproat and Shih 1991 for discussion.) It is often noted (e.g., Greenberg 1966; Cinque 1994) that adjective orderings generally occur in one of three possibilities:

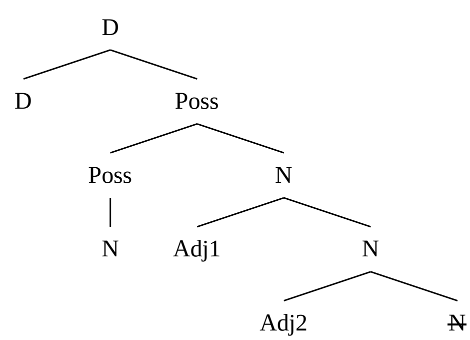
(26) a. D Adj1 Adj2 N (English)
b. D N Adj2 Adj1 (Arabic)
c. D N Adj1 Adj2 (Irish)
d. *D Adj2 Adj1 N (not attested)

That is, adjectives can be prenominal and postnominal, but only postnominal adjectives can have the so-called mirror image order that Arabic has. The so-called English order on the other hand can occur both prenominally and postnominally. In other words, there is a clear gap in the system: the fourth logical possibility, prenominal adjectives with mirror image order, as in (26d), is not attested.

We have seen in the previous section how RLin accounts for the adjective orderings in (26a) and (26b). However, we of course need to answer the question whether RLin can also account for the Irish order in (26c), and whether there is a principled way to rule out the unattested order of (26c).

The standard analysis of the Irish adjective order is N-movement (see, e.g., Sproat and Shih 1991) This analysis carries over to an RLin approach without modification:

(27)



If N moves to a higher head (say Poss) and if we assume an adjunct-first setting for N, the word order that is obtained is exactly the one in (26c): D N Adj1 Adj2.

Now what about the unattested order of (26d)? If N-movement combined with an adjunct-first setting yields postnominal adjectives with English order, we may wonder whether N-movement combined with an adjunct-second setting yields prenominal adjectives with an Arabic order. Interestingly, however, the order we derive in this case is the Arabic order in (26b), and crucially *not* the unattested one in (26d). We derive this order because N has moved up to Poss, and Poss will be linearized before its complement NP. As a result, N will appear before any lexical material remaining in the NP, which includes both adjectives. The adjunct-second setting on N will result in a mirror-image order for the adjectives, but they will be post-nominal.

The tree in (27), therefore, does not yield the unattested order under any parameter settings, it seems. Obviously, if we were to apply movement freely, some way could most likely be found to derive the order in (26d), but as long as there is no reason to assume those movements, we can conclude that under RLin the unattested order cannot be derived.

4.2. Movement as a leftward shifting operation

The deeper reason for the inability to derive the unattested order is that the Poss head to which N moves in our example has the default head parameter setting of H^+, given that it does not have an independent morphological form. It is this setting that causes Poss-N to be linearized before its complement, which contains the adjectives.

In this way, the proposal that H^+ is the default setting for the head parameter makes a strong prediction: movement of a head X to a head position Y will always result in X shifting leftward in the linear string. This is because the position from which X moves will always be in the complement of Y, and Y, not having an independent morphological form (if it did, nothing would move to it), will always be linearized before its complement.[30]

Note that this prediction about head movement is accompanied by a similar prediction about phrasal movement, i.e., movement to a non-head position. Principle P' states that a non-projecting element must be linearized first in its node. As a result a moved element will always be linearized before its sister, which contains the source position of the moved element.[31]

[30]Note that it is irrelevant whether X-to-Y movement results in substitution or in adjunction, because in both cases Y does not have an independent morphological form. How exactly the system would work out for head incorporation, however, needs to be worked out.

[31]For this, it is crucial that the target position of movement is not an adjoined position, which is, however, a standard assumption: movement is triggered by features, or, in Chomsky (2008), by the edge feature.

Combining these two predictions yields a more fundamental prediction about the language system: word order variation exists in positions created by external merge, but not in positions that are targets for movement.[32] If this prediction is correct, we also understand why there is no default setting for the adjunct parameter. Adjuncts are externally merged rather than moved into their adjoined position. Therefore, the null hypothesis is that there is word order variation in adjunct placement, and in the current model this can be captured by stating that the adjunct parameter does not have a default value.[33]

These observations of course raise the question why. Why is it that movement always yields a shift forward? The theory-internal reason is of course that movement is the result of internal merge triggered by some feature, so that it is subject to P'. Speculating a bit, however, we might say that parsing requires movement to be reconstructable; that is, a hearer, who only receives a linear string, without any indication of the hierarchical structure, must be able to reconstruct which elements have been moved, and from where. That is, the parser needs to be able to reconstruct the source position of a moved element, and presumably mark it.

Under a copy theory of movement, we can safely assume that marking of the source position is done simply by placing a copy of the moved element in that position.[34] If movement had the effect that the moved element appears further backward in the linear string, i.e., that the source position precedes the target position, the parser would have a major problem: it would not be able to determine of which element it must place a copy in the source position, simply because the target position (and hence the moved element) has not been parsed yet.[35] As a result, the parser would often have to retrace its steps, erasing and rebuilding large parts of its tree, especially in cases where it is not immediately clear that movement has

[32]Rightward movement is sometimes argued to occur in cases such as heavy-NP shift, right-dislocation and extraposition. The current model would require a non-movement analysis for these phenomena, which does not seem unfeasible. Cf. footnote 17.

[33]Under the assumption, that is, that adjuncts can adjoin to heads without independent morphological form. If this is the case, a default setting for the adjunct parameter would fix those adjuncts, disabling word order variation. An alternative way of capturing the insight would be to state that adjuncts only adjoin to heads with independent morphological content, which by definition can set their own linearization parameters. It is not entirely clear, though, whether this would be a feasible formulation, so I will opt for the one in the text.

[34]The assumption being that the parser will reconstruct a syntactic tree that is identical to a tree created by the computational system. Ideally, even, the parser uses the same mechanism used by the computational system, i.e., Merge. See Mulders (2002) for a discussion and implementation of this idea.

[35]At first sight, one might think that in a rightward shifting system, it would also be difficult for the parser to determine that movement has taken place at all. This is probably the case in some configurations, but certainly not in all. Also note that in the existing, leftward shifting, system, this is sometimes equally problematic, e.g. subject movement in SVO languages is not always obvious.

taken place. Therefore, a system in which movement yields a forward shift is more economical on the parser.

The reader will notice that there is a discrepancy here: the parser requires moved (i.e., internally merged) elements to be shifted leftward, but the mechanism responsible for the (apparent) leftward shift is principle P'. The point is that this principle applies not only to internally merged elements, but also to externally merged elements (at least those that are merged to satisfy some selectional feature). It could be the case, however, that the first occurrence of an internally merged item (i.e., the occurrence in the highest position) is not identifiable for RLin as an internally merged item. If so, the non-projecting nature of the element is the only available criterion that will positively identify all internally merged elements: it will yield some false positives, but it will not yield false negatives. In other words, principle P' is a solution to a restriction of the SM interface. It may not be a perfect solution, because of the false positives it yields, but it is probably the optimal solution: it is effective and very simple. As such, principle P' satisfies the Strong Minimalist Thesis.

The default value H^+ of the head parameter can be motivated in the same way: a head that moves must shift leftward, for the same reason that moved phrases must shift leftward. In order to achieve this, target positions for head movements must be linearized before their complements, something that is achieved easily by the default value H^+.

5. RLin and the LCA

The system outlined in this article differs considerably from the antisymmetric approach developed by Kayne (1994), which has been very influential over the past decade. It may therefore be worth our while to look at the two systems side by side. In essence, the LCA is more restrictive in the basic orders it allows (it really only allows one: Spec-Head-Comp), but as a result must assume large-scale movement operations and undesignated functional heads. With RLin, matters are more or less reversed: it is more permissive in the basic orderings it allows, but for that reason does not need to resort to movement operations to account for alternative orderings in the way that the LCA does.

The LCA has its basis in the Government and Binding framework, which is clear from the fact that it attempts to derive one of the central notions of G&B, X-bar theory, from more basic concepts.[36] Indeed, the ability to derive X-bar theory is

[36]This is not to say that it is non-minimalist. While the LCA may not have been conceived with the minimalist program in mind, it can nonetheless be seen as typically minimalist, in the sense that

one of the important theoretical arguments in favor of the LCA.

There is a second important, less theory-internal, theoretical claim that the LCA makes. The LCA is based on the idea that word order is a fundamental property of language, in that it derives directly from two basic concepts, c-command and dominance. In this way, word order is tightly integrated with core syntactic computation and all word order variation directly reflects hierarchical structure. In other words: there can be no such thing as language without word (or rather, constituent) order.

The intuitive appeal of such an approach is that it offers a very restrictive theory: the theory itself causes certain analyses to be impossible, and thus helps to account for the data we observe. One simple example can be found in the OV/VO variation. In principle, there are two ways we can account for this: 1) either we say that both orders are base-generated; or 2) we can say that one is derived from the other. With the LCA, we no longer face this choice: since the LCA restricts base-generation to VO orders, we conclude that option 1 cannot be correct, and furthermore we conclude that it is the OV order that is derived from the VO order, not the other way around. As such, the restrictiveness of the theory guides our analysis.

RLin, the approach discussed in the current article, is based on a different implementation of the grammatical framework, and more importantly, on a rather different conception of the relation between word order and core syntax. RLin incorporates the idea that word order is determined at PF: in core syntax, only hierarchical relations are relevant (assuming a compositional semantics), and word order is derived in a separate component. Put differently: the computational system produces an order-less I-language, and it would be in principle possible that E-language, language as we see it, were order-less as well, but for the fact that language needs to be expressed in some medium, and that this medium (be it speech or sign) imposes word order. In contrast, an LCA-based approach would claim that the computational system cannot, as it stands, produce language that is order-less.

This does not mean that operations in core syntax cannot have any influence on word order (as we have seen in the discussion above about the position of the possessor noun in the Saxon genitive), but it does mean that word order cannot be taken as a direct reflection of hierarchical relations. Instead, we will need to rely on other tests to establish the hierarchical relations in a phrase (i.e., binding phenomena, theta or case relations, cross-over, etc.)

Although it is not impossible to reformulate the LCA in terms of bare phrase structure (as Chomsky 1995 does, for example), doing so changes nothing about the assumption that word order is essentially determined in core syntax. Under

it tries to reduce phrase structure theory to more primitive notions.

such a reformulation, it is still the case that hierarchical relations directly determine word order, and therefore core syntactic operations still need to set up the tree in such a way that the correct word order can be derived.

Therefore, if one wishes to assume that word order derivation is purely a PF phenomenon and that hierarchical structure does not necessarily reflect word order directly, the adoption of a system similar to RLin seems inevitable. Furthermore, it is also inevitable that such a system be more permissive than the LCA, for the simple reason that it cannot *move* constituents in order to derive the correct word order.[37] It only has the parameter settings in order to account for word order variation.

To take up the above example of OV/VO variation again: if we wish to account for it in an RLin approach, we must assume that both OV and VO orders can be "base-generated".[38] As such, an RLin-based approach is necessarily more permissive. While an antisymmetric approach excludes OV as a basic order, RLin allows it.

It should be noted, however, that although RLin is more permissive than the LCA, it cannot be said that it is unrestricted. The most fundamental restriction is found in the manner in which a tree is traversed: RLin must necessarily start at the root node, and must complete a branch in its entirety before it can continue with its sister branch. This is a direct result of the recursive nature of the depth-first search adopted for RLin, and it puts a strong restriction on the linear orders that can theoretically be derived from a given tree.

Furthermore, it should be kept in mind that a restriction in one part of the theory usually has a trade-off in another part. We can demonstrate this for the LCA if we look at the universal adjective orderings discussed in the previous section again. In general, an antisymmetric approach can handle the relevant data rather elegantly, although as will be demonstrated, there is a trade-off.

First, because of the Spec-Head-Comp order, the LCA determines that the English order in (26a) is the basic order. From the basic order we can easily derive the Irish order in (26c) by applying N movement, leaving the adjectives in place

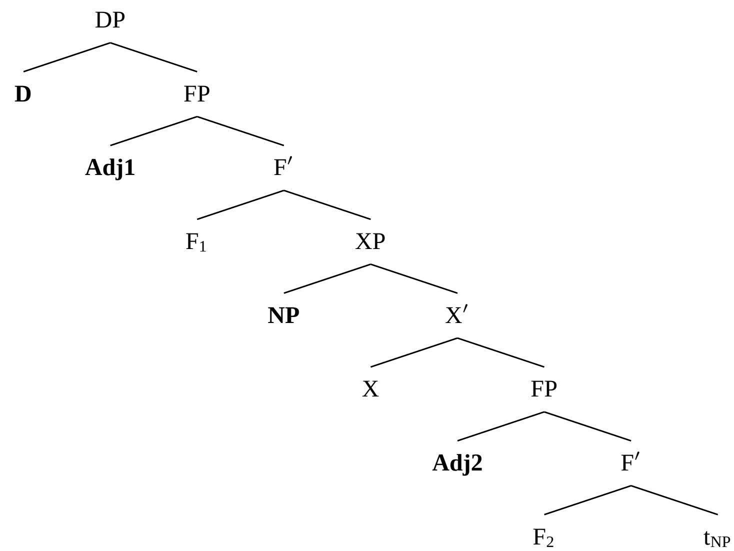
[37] At least, on the assumption that there is no such thing as PF movement. I will leave aside the question whether head movement is (partially) a PF-phenomenon.

[38] "Base-generated" not being the entirely appropriate term, of course, because the base-generated structure itself is order-less. What is meant is that both orders are derived in the same manner, without one being derived from the other. Note, by the way, that this does not imply that both orders must be exact mirror images of each other. Neeleman and Weerman (1999) show that differences between VO and OV languages (e.g. the fact that OV languages generally allow scrambling, while VO languages do not) can be accounted for under a base-generation analysis if we assume that the case licensing domains correlates with word order: in VO structures the licensing domain is defined phonologically, while in OV structures it is defined syntactically, which yields a somewhat larger domain.

but moving N to a higher and therefore more anterior position.[39]

The mirror image order in (26b) can be derived through so-called *roll-up*, as in Cinque (1994). Assume that the adjectives are in specifier positions of some functional head, which we shall call F.[40] In the first step of roll-up, the NP moves to a specifier position of some projection immediately above the lower adjective, say Spec,XP, yielding an intermediate tree (28) (bold face indicates overt material):

(28)

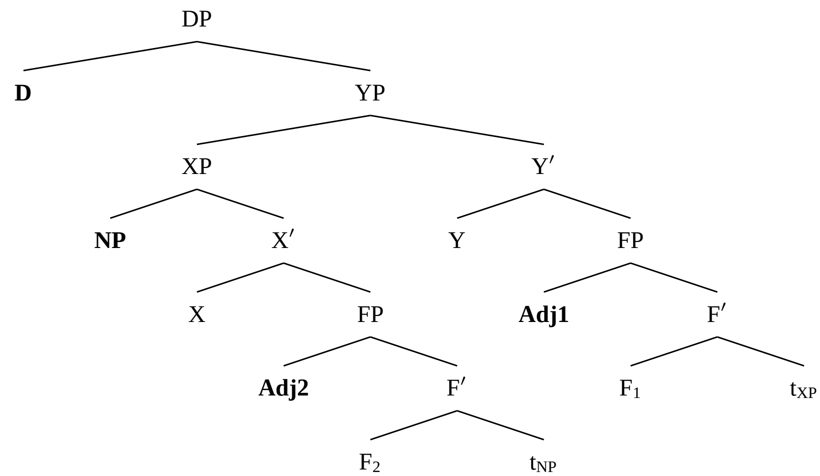


After this, XP moves to a position above the higher adjective, say Spec,YP. This movement moves not only NP but also Adj2 (and t_{NP}, so it is remnant movement), yielding (29), which gives the desired order:

[39] Or, if one disallows head-movement, through movement of NP to a high position.

[40] This assumption is necessary, because the LCA does not allow multiple adjunction. Although note that under Chomsky's reinterpretation, with the LCA operating at PF, multiple adjunction does seem possible.

(29)



Crucially, there is no way to derive the fourth possibility. If we apply roll-up in combination with N or NP movement, the adjectives will have changed order, but N will still be higher. Therefore, this derivation yields not (26d) but rather (26b). So we see that the restrictiveness in the LCA actually helps us to account for the fact that a certain word order is unattested.

Yet, we also see that the restrictiveness has its trade-off, in that the theory needs to be more permissive in the movements it allows. The roll-up movements demonstrated here are rather powerful, and do not seem to be directly motivated in terms of features; at least, Cinque (1994) and Shlonsky (2000) do not make clear which features are thought to trigger these movements.

Note also that crucially, we can only adopt this as our account of the absence of (26d) if we place appropriate restrictions on possible movements. It is not possible to simply allow any kind of movement to take place, because that would enable us to derive the non-existent order rather easily, e.g., by moving Adj2 to a specifier position above Adj1, say Spec,XP (essentially roll-up without including the NP). However, LCA-style analyses generally make no attempt to formulate a principled proposal on the kinds of movement that an antisymmetric framework should allow, and which restrictions should be put on it.[41]

With RLin, things are more or less reversed. The basic theory is itself more permissive than the LCA, but this means that movement can be more restricted. As we have seen in the previous sections, no movements are needed in order to derive the mirror image order for adjectives. In fact, in the analysis of the given empirical domain, we are able to restrict movement to only those cases where it is empirically supported, specifically, the movement of the possessor to Spec,D/PossP in English.

[41]Which, in my opinion, seriously undermines their explanative power.

6. Summary and conclusions

The goal of this article has been to investigate what a linearization procedure that operates at PF might look like, and what it would give us. A PF linearization procedure must search the syntactic tree for terminal elements. The null hypothesis is that the procedure will spell out an element as soon as it finds it, where spell-out is to be understood as "to put in the linear string".[42]

The most efficient way to search the kind of binary trees that form linguistic structures is through a recursive search procedure that starts at the root node. The only such procedure that is compatible with standard syntactic assumptions is the so-called depth-first search, which searches a tree branch by branch. As a result of this search procedure, elements that are sisters in the tree will end up adjacent in the linear string.

This, however, is not sufficient for linearization. In each node, the procedure, which is dubbed RLin for short, needs to determine which of the two sisters to linearize first. In order to allow RLin to make this decision, one principle and two parameters are proposed:

- Principle P': Linearize the non-projecting subnode first.
- Head parameter:

  H^+ : Linearize a head immediately.

  H^- : Do not linearize a head immediately.
- Adjunct parameter: adjuncts are linearized first or second in their node.

In principle, the two parameters can be set for each category independently. It seems plausible, however, both from an acquisition and from a theoretical perspective, that only categories that have an independent morphological form can do so. Categories that do not will adopt the default H^+ setting for the head parameter, and simply copy the setting of their complement for the adjunct parameter (or may not have a value for it at all).

With the system thus outlined, it is possible to account for the word order in a subset of the Arabic and English noun phrases, while maintaining an almost identical structural description for the noun phrase in both languages. The only structural

[42] Although the null hypothesis, it is tempting to consider the possibility that RLin may under certain conditions postpone spell-out of some element, e.g. a phase, when it is "heavy", by some phonological measure. It may be possible to develop an account for heavy-NP shift along these lines, as suggested in footnote 17.

difference between the two languages in the domain discussed is the fact that in English the possessor in a Saxon genitive construction moves to Spec,D/PossP, while in Arabic the possessor in a construct state remains in its base position in Comp,NP.

RLin makes a strong claim about word order: word order variation only exists in positions created by external merge. Elements that move, either heads or phrases, are always linearized first. As a result, movement of an element X(P) generally results in a shift of X(P) to a more anterior position in the linear string. The deeper reason behind this fact may be found in SMT: it may be the optimal solution to parsing restrictions imposed by the SM system.

References

Abney, Steven (1987). The english noun phrase in its sentential aspect. Ph.D. Dissertation, MIT, MIT Working Papers in Linguistics.

Ackema, Peter and Ad Neeleman (2004). *Beyond Morphology: Interface Conditions on Word Formation*. Oxford: Oxford University Press.

Borer, Hagit (1999). Deconstructing the construct. In *Beyond Principles and Parameters*, Kyle Johnson and Ian Roberts (eds.), 43–90, Dordrecht: Kluwer Academic Publishers.

Bratko, Ivan (2001). *Prolog Programming for Artificial Intelligence, 3rd ed.* Harlow: Addison-Wesley.

Chomsky, Noam (1995). *The Minimalist Program*. Cambridge, Mass: The MIT Press.

Chomsky, Noam (2004). Beyond explanatory adequacy. In *Structures and Beyond*, Adriana Belletti (ed.), vol. 3 of *The Cartography of Syntactic Structures*, 104–131, Oxford: Oxford University Press.

Chomsky, Noam (2008). On phases. In *Foundational Issues in Linguistic Theory*, Robert Freidin, Carlos Otero and Maria-Luisa Zubizaretta (eds.), Cambridge, Mass: The MIT Press.

Cinque, Guglielmo (1994). On the evidence for partial N-movement in the Romance DP. In *Paths Towards Universal Grammar: Essays in Honor of Richard S. Kayne*, Guglielmo Cinque, Jan Koster, Jean-Yves Pollock and Rafaella Zanuttini (eds.), Washington, D.C.: Georgetown University Press.

Fassi Fehri, Abdelkader (1993). *Issues in the Structure of Arabic Clauses and Words*, vol. 29 of *Studies in Natural Language and Linguistic Theory*. Dordrecht: Kluwer Academic Publishers.

Giorgi, Alessandra and Fabio Pianesi (1997). *Tense and Aspect: From Semantics to Morphosyntax*. New York/Oxford: Oxford University Press.

Greenberg, Joseph Harold (1966). Some universals of grammar with particular reference to the order of meaningful elements. In *Universals of Language*, Joseph Harold Greenberg (ed.), 73–113, Cambridge, Mass: The MIT Press, 2nd ed.

Grimshaw, Jane (1990). *Argument Structure*. Cambridge, Mass: The MIT Press.

Halle, Moris and Alec Marantz (1993). Distributed morphology. In *The View from Building 20: Essays in Linguistics in Honor of Sylvain Bromberger*, Kenneth Hale and Samuel Jay Keyser (eds.), vol. 24 of *Current Studies in Linguistics*, chap. 3, 111–176, Cambridge, Mass: The MIT Press.

Higginbotham, James (1985). On semantics. *Linguistic Inquiry*, 16(4):547–539.

Holm, John (2000). *An Introduction to Pidgins and Creoles*. Cambridge: Cambridge University Press.

Kayne, Richard (1993). Toward a modular theory of auxiliary selection. *Studia Linguistica*, 47:3–31.

Kayne, Richard (1994). *The Antisymmetry of Syntax*. Cambridge, Mass: The MIT Press.

Kremers, Joost (2003). The noun phrase in Arabic: A minimalist approach. Ph.D. Dissertation, University of Nijmegen, LOT Dissertation Series 79.

Laenzlinger, Christopher (2000). French adjective orderings: Perspectives on DP-internal movement types. *GG@G*, 1:55–104.

Lebeaux, David (1991). Relative clauses, licensing, and the nature of the derivation. In *Perspectives on Phrase Structure: Heads and Licensing*, Susan D. Rothstein (ed.), vol. 25 of *Syntax and Semantics*, San Diego: Academic Press.

Longobardi, Guiseppe (2001). The structure of DPs: Some principles, parameters and problems. In *The Handbook of Contemporary Syntactic Theory*, Mark Baltin and Chris Collins (eds.), 562–603, Malden, Mass./Oxford: Blackwell Publishers.

Moro, Andrea (2000). *Dynamic Antisymmetry*. Cambridge, Mass: The MIT Press.

Mulders, Iris (2002). Transparent parsing: Head-driven processing of verb-final structures. Ph.D. Dissertation, Utrecht School of Linguistics UiL-OTS.

Neeleman, Ad and Fred Weerman (1999). *Flexible Syntax: A Theory of Case and Arguments*. Dordrecht: Kluwer Academic Publishers.

Reuland, Eric (1990). Head movement and the relation between syntax and morphology. In *Yearbook of morphology 3*, Dordrecht: Foris.

Ritter, Elizabeth (1991). Two functional categories in noun phrases: Evidence from Modern Hebrew. In *Perspectives on Phrase Structure: Heads and Licensing*, Susan D. Rothstein, (ed.), vol. 25 of *Syntax and Semantics*, 37–62, San Diego: Academic Press.

Shlonsky, Ur (2000). The form of the Semitic noun phrase: An antisymmetric, non N-movement account, ms. University of Geneva.

Sproat, Richard and Chilin Shih (1991). The cross-linguistic distribution of adjective ordering restrictions. In *Interdisciplinary Approaches to Language. Essays in Honor of S. Y. Kuroda*, C. Georgopolous and R. Ishahara (eds.), 565–593, Dordrecht: Kluwer Academic Publishers.

Stepanov, Arthur (2001). Late adjunction and minimalist phrase structure. *Syntax*, 4(2):94–125.

Szabolcsi, Anna (1994). The noun phrase. In *The Syntactic Structure Of Hungarian*, Ferenc Kiefer and Katalin Kiss (eds.), vol. 27 of *Syntax and Semantics*, 197–174, San Diego: Academic Press.

Winston, Patrick (1992). *Artificial Intelligence, 3rd ed.* Reading, MA: Addison-Wesley.

Zribi-Hertz, Anne (1997). On the dual nature of the 'possessive' marker in Modern English. *Journal of Linguistics*, 33(2):511–538.

Zwarts, Joost (1992). X′-syntax – X′-semantics: On the interpretation of functional and lexical heads. Ph.D. Dissertation, Utrecht School of Linguistics UiL-OTS.