More Multiple Multiple Spellout

Meaghan Fowlie

March 19, 2010

1 Introduction

The purpose of this paper is to correct and extend the research program laid out in Fowlie (to appear). The previous version of Multiple Multiple Spellout focuses on free word order languages; here we will take a closer look at strict word-order languages. In particular, we will examine the consequences of Nunes and Uriagereka (2000)'s version of the LCA and spellout.

Section 2 will provide a brief overview of Fowlie (to appear). The next section will be a discussion of PIC and CED effects predicted by the theory. Next we will look at predictions about movement made by the theory: that only phases can move in strict word-order languages. Finally, we will examine some particular cases of specifiers, and how they behave in the theory.

The major prediction of interest is this: only constituents that have been spelled out can be specifiers and adjuncts. Moreover, since all movement is movement to specifiers or adjunct positions, only constituents that can be spelled out can move at all.

The two interpretations of this prediction are the following: One, to start with the claim that only phases can be spelled out. This entails that PPs are phases. It also predicts that only phases can be remnant-moving, restricting the possibilities of remnant movement considerably. Two, to claim that movement causes spellout. This latter claim turns out to be stranger than it seems, within the framework used here. It will turn out to predict that whether or not the numeration can contain non-trivially structured elements is a parameter, not a principle.

2 Background

Fowlie (2007) proposes that Spellout is in fact two separate operations: one which defines a linear order but does not render the domain opaque to movement (LINEARISE), and one which sends phonological features to the phonological component and does render the domain opaque (ATOMISE).
LINEARISE is based on Fox and Pesetsky (2005)’s Linearize, with the ordering algorithm specified as a version of the Linear Correspondence Axiom (Kayne, 1994) proposed by Nunes and Uriagereka (2000).

**Linear Correspondence Axiom (Kayne, 1994):** For any pair of non-terminal nodes \( <X, Y> \), if \( X \) asymmetrically c-commands \( Y \) then each terminal node dominated by \( X \) precedes each terminal node dominated by \( Y \). Moreover, the set of all such correspondences constitutes a total ordering on the terminal nodes.

Kayne assumes that the terminal nodes (e.g. lexical items) project up to a syntactic head without branching. This assumption is necessary to derive a total ordering on the terminal nodes.

Conversely, Nunes and Uriagereka (2000) propose that the Minimalist assumption of Bare Phrase Structure is correct (i.e. terminal nodes do not project up to a syntactic head without branching) and that the LCA is simpler than Kayne’s statement. In particular, they remove the notion of dominance from the LCA.

**Linear Correspondence Axiom (N&U 2000):** A Lexical Item \( \alpha \) precedes a Lexical Item \( \beta \) iff \( \alpha \) asymmetrically c-commands \( \beta \).

In order to derive a total ordering on the terminals, N&U propose that specifiers and adjuncts are spelled out before being attached to the tree. This spellout not only renders the constituents opaque to the derivation; it completely removes the structure from the derivation, leaving behind a single terminal element to mark the place of the spelled-out constituent.\(^1\) This terminal element can now be treated like a Lexical Item by the LCA, and be linearised with the rest of the structure. Without this “packaging up” of the specifier, there is no c-command relationship between the lexical items of the specifier and the rest of the tree, since they are too embedded.

\(^1\)See Fowlie (to appear) for a discussion of the status of this removed constituent. The upshot is that either the constituent just *seems* to be removed, but is in fact still there so that LF can receive its structure at the end of the derivation, or there is LF spellout so that LF can receive its structure as the derivation goes along.
Lexical Items of a complex specifier have no c-command relationship with Lexical Items of \V'

A Spelled-out specifier is a terminal node and asymmetrically c-commands the Lexical Items of \V'

I proposed that this spellout operation that renders the constituent opaque is a spellout operation in addition to F&P’s Linearize. I called it ATOMISE, since the derivation treats the spelled-out constituent as an atom.

**The two operations:**

**LINEARISE:** N&U’s LCA determines the linear order of terminals in the spellout domain. The order is stored on an ordering table; any contradictions will crash the derivation.

**ATOMISE:** The constituent is sent to PF. It is no longer accessible to the derivation.

The spellout domains (SD) are claimed to be universal across languages. What is parameterised is whether each of these operations occurs at each SD. A language is specified, for each SD, whether or not ATOMISE occurs and whether or not LINEARISE occurs. This
defines a typology. Fowlie (2007) tentatively proposes that CP, TP, VP, DP, and NP are spellout domains. The following model is proposed:

1. C – CP (Including the whole left periphery)
2. T – sister of C (approximately TP)
3. V – vP
4. D – DP
5. N – NP

Let \( [\pm L \pm A] \) notate whether or not linearise (L) and atomise (A) occurs. Then each type has the following form

\[
C[\pm L\pm A] \rightarrow T[\pm L\pm A] \rightarrow V[\pm L\pm A] \rightarrow D[\pm L\pm A] \rightarrow N[\pm L\pm A]
\]

For example, Type \( C[-L+A] \rightarrow T[-L-A] \rightarrow V[+L-A] \rightarrow D[-L+A] \rightarrow N[-L-A] \) has un-linearised but atomised DPs, linearised but un-atomised vP and un-linearised but atomised CP. No Spellout operations take place at TP or NP.

An SD that is atomised but not linearised has no order specified for it, and so will show free word order. See Fowlie (to appear) for details. Important for our purposes is that for some languages, such as Dyirbal (Dixon, 1972), satellites do not appear to be atomised: noun markers and adjectives can be separated from their nouns; in fact, within a clause, words can appear in any order at all. This can be accounted for if satellites are not atomised before being merged to the spine. In this case, the LIs of the satellite are not treated as a unit and so can appear anywhere in the order.

### 3 CED and PIC effects

Importantly, it is proposed that linearise occurs as soon as the SD is built, but atomise waits until the next phase is built. If linearise occurred one phase later, we would not see the effects of the order within VP: there would be too much time for elements to move out of VP before it was linearised. If atomise occurred immediately, nothing would be able to escape phases, not even the edge.

Fox and Pesetsky (2005) derive Phase Impenetrability (PIC) effects with their Linearize. Moreover, they derive the weakness of PIC effects. If PIC is taken as stipulation, the claim is that only the head and specifier of the phase edge can escape the phase. F&P predict rather that more than the head and specifier may move as long as linear order is retained. Therefore the PIC is a weak epiphenomenon of Linearize rather than an absolute.

The Condition on Extraction Domains, on the other hand, is that there is an asymmetry between specifiers and adjuncts on the one hand and complements on the other. There is no escape hatch for movement from specifiers and adjuncts. This is not the same effect as the PIC.

With a little tweaking, Fowlie (to appear) can derive this distinction. Anything that is linearised but not (yet) atomised will show PIC effects. Since the complement is part
of the main spine of the tree, it acts like VP in F&P’s proposal: you can escape it as long as you do not disrupt the linear order. CED effects, on the other hand, are the effect of ATOMISE: specifiers and adjuncts must be ATOMISED before they can attach to the tree; otherwise no linear order between them and the rest of the tree can be derived.

The CED effects Multiple Multiple Spellout derives are the result of a necessary stipulation: that while ATOMISE normally waits until the end of the next phase, when the phase is built in a separate derivational workspace – i.e. when it is a specifier or adjunct – it must be ATOMISED right after LINEARISE; otherwise there is no “next” phase to trigger ATOMISE. If the specifier or adjunct is to be linearisable with the rest of the structure, it must be ATOMISED.

How can we guarantee that ATOMISE will occur at this juncture? There are two possibilities:

1. Only phases can be satellites. If this is the case, we stipulate that if a phase P is marked $P[+A]$ it is ATOMISED when the workspace is finished with. This may seem as though it can be unified with the general fact that the whole sentence must be ATOMISED even though it is not embedded in another phase. Unfortunately, it cannot be the same trigger: languages that do not ATOMISE their satellites, like Dyirbal, nonetheless do ATOMISE the whole sentence, or else nothing would be pronounced, as PF would get nothing at all. (Dyirbal is not the null language!)

2. More than just phases can be satellites. In this case, in order to derive an ordering between satellites and the spine, satellites must be spelled out whether or not they are phases. We need a new trigger for spelling out constituents: a constituent C is spelled out when its workspace is finished with, whether or not C is a phase. The spellout is triggered not by the specification of the phase, but procedurally: the workspace is finished with, so the result is spelled out.

This trigger must be parameterised in order to account for the case of Dyirbal discussed above: some languages do not ATOMISE their satellites.

We add to the typology a special parameter that determines whether or not a constituent is spelled out when a workspace is finished with: $W[±A]$.

It turns out this parameterisation can account for Hindi Subject Condition violations such as that in (3)$^2$

(3) Kiskii tum socte ho ki $[SUB.twh \text{kitaab}]$ corii ho gayii whose you think that $[SUB.twh \text{book}]$ stolen was ‘Whose book do you think was stolen?’

Hindi specifiers are strictly ordered with the rest of the sentence: scrambling them causes binding changes. They are therefore ATOMISED so that they can LINEARISE with

$^2$Data from Mahajan (1992)
the spine. However, they are not islands. We can account for this if Hindi DPs are [+A] but Hindi is $W[-A]$. 

In this case, the DP *kiskii kitaab* ‘whose book’ is LINEARISED when it is built. The workspace is now finished with, but Hindi is $W[-A]$, so it is not ATOMISED. Next, the DP is merged with the spine in the main workspace. Since the structure is still there, *kiskii* ‘whose’ is still free to move out. When the next phase in the main workspace is complete, any subphases that are specified [+A] are ATOMISED, including $[DPt_{with} kitaab]$. Now LINEARISE applies, and thanks to DP being specified [+A], the DP is linearisable with the spine.

These two solutions are combinable as well. The parameter $W[±A]$ can still exist even if only phases are ever spelled out. This is required to account for the Hindi data even if only phases can be spelled out.

The CED predictions of Multiple Multiple Spellout rest on the stipulation that a constituent can be ATOMISED when its workspace is finished with. CED effects do not fall out of the definitions of ATOMISE and LINEARISE by themselves. There is still a certain amount of intuitive appeal, however. The idea that when a workspace is finished with, the result is spelled out makes sense procedurally, and parallels the fact that the whole sentence must be spelled out when it is finished being built too.

4 Movement

Recall that in languages with strict order, all satellites must be ATOMISED in order to derive a total ordering on the terminal nodes. Where do satellites come from? They have two origins. The first origin is separate derivational workspaces. Anything that starts its life as a satellite is built separately. It is ATOMISED by virtue of the language being $W[±A]$ or it is ATOMISED by virtue of being a phase.\(^3\)

The second origin is movement of part of the spine. This can happen with movement of or out of the complement of the verb. It can also happen with remnant movement. Somehow these moved elements need to be ATOMISED.

There are two basic ways to think about this, each with their own implications. One, we take this to define phases and restrict remnant movement. Two, there is something that forces ATOMISE of anything that moves, leaving remnant movement unrestricted. This second take on movement makes it difficult but not impossible to derive the Hindi and Dyirbal facts, as we will see. Let us consider each of these in turn.

4.1 Take one: movement is restricted to phases

First let us make the problem clear. Consider the following from English and Dutch\(^4\):

\(^3\)Note that I predict that Hindi can only build phases in separate workspaces. If non-phase satellites are built separately and then merged, they will have no reason to ATOMISE since Hindi is $W[-A]$. Further research will show whether this prediction is borne out.

\(^4\)Dutch data from Floris van Vugt, p.c. 2010
We can easily account for the English datum. It is possible to move out of the direct object because it is part of the spine. English is $W[+A]$ but the object is not built in a separate workspace, so the wh-phrase can move out. Moreover, even though whose mother is complex, it can be linearised with the rest of the sentence because DP is a phase. It will be atomised after the DP containing it is built, which is certainly before the first phase to contain it as a specifier is linearised.

Dutch, however, is different. Notice the pied piping: the wh-phrase is not a DP; it is a PP. If PP is not a phase, how can we account for this?

Under this account, we take the fact that PP can be a specifier as proof that it is a phase after all. If we accept Take One, we must accept PP as a phase.

This approach also predicts that only phases can remnant-move. It is possible to derive many (or maybe any) word-orders in a Kaynian framework as long as we can find enough phrases to move things into, and as long as we can remnant-move whatever we want. This approach restricts that. Only phases can move, not just any phrase. Further research will show whether this is an accurate prediction.

4.2 Take Two: movement forces ATOMISE

The second take on this is to allow anything to move, but to force spellout when things move. A clear way to look at this is through the Numeration. Suppose we claim that each phase has its own numeration, or set of things from which things can be merged. Suppose further that the only things that can be merged pairs from the numeration or one thing from the numeration and the tree in a workspace. In this case, you can never merge two trees from two workspaces. One of the trees must come from the numeration.

It is not unreasonable to suggest that something can only be added to the numeration if it has no internal structure. Only LIs, and things that look like LIs (i.e. labels of atomised constituents) can be added to the numeration. What does this predict?

If you build a specifier in a separate workspace and want to merge it to the tree, this can only be done by atomising the specifier, and then adding the atom back into the numeration. Now it can be merged with the tree.

Similarly, Move works by putting an already built structure into the numeration. For example, if we want to do vP-remnant movement, vP is added back into the numeration, then re-merged later. Again, something can only be added to the numeration if it has no internal structure, so vP must be atomised before it is added to the numeration.

What we have now is that instead of $W[±A]$, specifiers are spelled out because of a requirement of the numeration: it cannot contain trees, only potential terminal nodes.

This seems rather nice, but now we cannot account for Dyirbal and Hindi, wherein satellites are not atomised before being merged to the spine. We must then say that this
requirement on the numeration is parametereised: some languages allow non-trivial trees in their numerations, and some do not. The question then arises: should something this fundamental be a parameter? Should it not be a principle?

5 More specifiers

This section will look at objects in specifier positions, and what they mean for CED effects. We will look at the difference between indirect and direct objects, and at the behaviour of direct objects in OV languages like German and Dutch.

5.1 Indirect Objects

CED effects distinguish between subjects/adjuncts and objects. However, with binary branching there is a syntactic difference between direct and indirect objects. If CED effects are a syntactic phenomenon, one would expect that one of these two objects types should also be an island, since the verb can only have one complement.

As it turns out, indirect objects show subject-like behaviour (CITE). Consider the following English data.

(5) a. I showed [IO a picture of Dumbledore] [DO a picture of Sirius].
   b. Who did you show [IO a picture of Dumbledore] [DO a picture of t]?
   c. *Who did you show [IO a picture of t] [DO a picture of Sirius]?

This is a general phenomenon: indirect objects pattern not with direct objects and subjects of unaccusatives, but with subjects of unergatives.\(^5\) If the former are underlying complements and the latter underlying specifiers, this approach accounts for their behaviour.

These data predict the following double object construction:

(6)\[ 
\begin{array}{c}
  \text{vP} \\
  \text{SUB} \\
  \text{v} \\
  \text{VP} \\
  \text{IO} \\
  \text{V} \\
  \text{DO} \\
\end{array}
\]

The indirect object is an underlying specifier, and so should and do show subject-like behaviour with respect to extraction.

5.2 OV languages

Under the LCA, languages like Dutch and German have their OV order derived by movement. The direct object starts as a complement but moves to a specifier position above the verb.

\(^5\)Hilde Koopman, p.c. 2010
Now we must ask how the derived complement linearises with V. It needs to be atomised in order to linearise with V. Consider the following Dutch datum:

(8) \[\text{Van wie} \] heb je \[\text{DO een foto t} \] gezien?
    \[\text{Of whom} \] have you \[\text{DO a picture t} \] seen?
    ‘Who did you see a picture of?’

\text{Van wie} ‘of whom’ is easily extracted from the direct object, so it is not an island, despite supposedly being a specifier.

DP is a phase, so when VP is finished being built it gets spelled out even if it has already moved to a specifier.

To illustrate, let us build the sentence above. The object moves into some specifier, say of \text{Arg}_O. Now it is to the left of the verb.

(9)

The object still has internal structure (it hasn’t yet been atomised). The wh-phrase can move out to the left edge of whatever the next phase is (TP?).
Now the phase is finished, so anything inside that is [+A] is ATOMISED. If PP is a phase, it is ATOMISED.

If PP is not a phase, and we want to go with Take Two (that movement forces ATOMISE), we are in trouble. DP will have been ATOMISED already, when it moved to spec-ArgO, so PP cannot escape. If we posit enough functional projections, we can find a spot for PP to have already moved into. Of course, it will have had to move through spec-DP too, since DP is a phase, and is [+L]. PP surfaces to the left of the rest of the DP so it must have moved through spec-DP.
However, DP still needs to move to the left of the verb, so it moves now too. Now the order is DP > PP > V, but we want PP > DP > V so PP will have to move again before the phase ends. What are all these specifiers, and what motives their movement there? This is a very messy solution, lending credence to Take One as against Take Two.

We conclude then that PP is a phase. To summarise, it is possible to escape a direct object in an OV language because even though it is a specifier, it is not atomised until after it is moved to a specifier position.

## Conclusion

The unexpected subject-like behaviour of indirect objects can be explained if it is a specifier just like a subject is. The approach outlined here derives specifier and adjunct islands by the application of spellout. It is claimed that whether or not underlying satellites are islands is parameterised. It depends on whether constituents built in a separate derivational
workspace are ATOMISED before being merged to the rest of the sentence. Satellites that are underlyingly part of the spine, on the other hand, are claimed to be restricted to phases. Only phases can be remnant-moved. This allows the derived satellites to LINEARISE with their sisters.

Nothing has been said of satellites in free word order languages. Since I derive the CED from linearisation requirements, I predict that satellites that are in unlinearised phases are not islands. Further reearch is required to determine if this prediction is borne out.

References


