Multiple Spellout

Abstract

It has been posited that Spellout, like other syntactic operations, can occur more than once (Bresnan 1971)). I suggest that what we call Spellout is in fact at least two separate operations: at minimum, one which determines the linear order of Lexical Items (“LINEARISE”) and another which sends the phonological features of the Spelled-out constituent to the phonological component of the language faculty, rendering that constituent unavailable to the syntactic derivation (“ATOMISE”). The separate application of these operations can yield phenomena such as Holmberg’s Generalisation and other successive cyclicity effects (Fox & Pesetsky 2005) – LINEARISE without ATOMISE – and scrambling – ATOMISE without LINEARISE.

1 Introduction

In the Minimalist Program (Chomsky 1995), Spellout is the syntactic operation that delivers to the phonological component of the language faculty everything it needs in order to form an utterable sentence. This includes all of the phonological features of the terminal nodes - be they lexical items or morphemes - and the order in which they are to appear. Joan Bresnan (1971) introduced the notion of Multiple Spellout: that Spellout can occur several times throughout the derivation. Working within this framework, researchers have been asking what exactly Spellout is, and how a Spelled-out constituent behaves. Given that phonological features and linear order are quite different sorts of entities, I suggest that Spellout is in fact (at least) two operations: one that assigns linear order to the terminal nodes of a constituent, and one that sends phonological features to the phonological component.

---

1I would like to thank the many students and professors who have taken time to advise me on this project, in particular Jon Nissenbaum, Tobin Skinner, Raphael Mercado, and especially my thesis supervisor Lisa deMena Travis, who was also responsible for pointing out the possibilities this approach presents for scrambling.
2 Background

2.1 Linear Correspondence Axiom (LCA)

X-bar theory offers no mechanical algorithm to map hierarchical structure to the surface linear form of language. Any pair of sisters can be stipulated to appear in either order. In his 1994 monograph *The Antisymmetry of Syntax*, Richard Kayne proposed that linear order is in fact derivable from hierarchical structure. He showed that it was possible to derive X-bar assumptions from c-command relations. In particular, he proposed the Linear Correspondence Axiom (LCA):

**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 \( X \) preceded each terminal node dominated by \( Y \). Moreover, the set of all such correspondences constitutes a total ordering on the terminal nodes.

Kayne assumes irreflexive dominance and that the terminal nodes (e.g. lexical items) project up to a syntactic head without branching. At least one of these two assumptions is necessary to derive a total ordering on the terminal nodes.

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 definition of the LCA.

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

The removal of dominance from the definition of the LCA means that Nunes & Uriagereka are no longer interested in anything but terminal nodes. This is in contrast to Kayne, who uses mathematical relations among non-terminals to determine linear order of terminals. Nunes & Uriagereka determine linear order of terminals directly from the mathematical relations among the terminals themselves. We will see how this works in more detail in section 2.3 below.

2.2 Fox & Pesetsky

Fox & Pesetsky (2003, 2005) propose that Spellout fixes the relative order of the lexical items in a Spelled-out domain. At the end of the construction of each Spellout Domain (SD) \( D_i \),
1. The elements of $D_i$ are linearised by some linearisation algorithm

2. Resultant ordered pairs are stored on an *Ordering Table*

They call this operation *Linearize*. The rule of *Order Preservation* states that no information is ever lost from the Ordering Table.

When the next Spellout Domain $D_{i+1}$ is Linearized, it is Linearized with respect to the first element of the previously spelled-out SD $D_i$. “First” is defined by the previous application of Linearize. $^2$ After a Spellout Domain is Linearized, constituents thereof can still move out of the SD, but they cannot change their order. This derives effects such as the Holmberg Generalisation, including aspects that are rarely accounted for.

Fox & Pesetsky also derive other types of successive cyclicity. Consider, for example, English object wh-movement. The object wh-word in its theta-position is after the verb and subject. For example:

(1) $[CP \text{ what did } [VP <\text{what}> \text{ he read } <\text{what}>]]$

<table>
<thead>
<tr>
<th>Ordering Table (VP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>what $&lt;$ he</td>
</tr>
<tr>
<td>he $&lt;$ read</td>
</tr>
<tr>
<td>what $&lt;$ read</td>
</tr>
</tbody>
</table>

Table 1: Ordering Table (VP)

<table>
<thead>
<tr>
<th>Ordering Table (CP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>what $&lt;$ he</td>
</tr>
<tr>
<td>he $&lt;$ read</td>
</tr>
<tr>
<td>what $&lt;$ read</td>
</tr>
<tr>
<td>what $&lt;$ did</td>
</tr>
<tr>
<td>what $&lt;$ VP</td>
</tr>
</tbody>
</table>

Table 2: Ordering Table (CP)

$^2$Although it may seem like a lot of extra stipulation that Linearize can tell when something has been spelled out already, the details of Fox & Pesetsky’s proposal are more rigorous than what I have presented here. In particular, the operation Linearize is *always* looking at the “first” and “last” elements of a domain; the difference is that sometimes the domain in question is a single Lexical Item (or possibly morpheme) with a “beginning” and an “end” as expected. See their 2003 handout for details. (A monograph is forthcoming.)
In the final form of the sentence, *what* is before the verb. If *vP* is an SD, *what* must be before the verb when the *vP* is Linearized. Otherwise, the Ordering Table will store the information that *read* precedes *what*, and then when the next SD is Linearized, the Ordering Table will receive the information that *what* precedes *read*. This contradiction will crash the derivation.

In their 2003 handout, Fox & Pesetsky propose a possible linearisation algorithm that relies on stipulated ordering of heads, specifiers, and complements. A major advantage over Nunes & Uriagereka’s version of the LCA is that simple sisters can be linearised (Cf. section 2.3 below). Nevertheless, Fox & Pesetsky comment: “we suspect that all or most of our proposal could be reformulated if c-command, rather than sisterhood, were the central notion for the Laws of Precedence (as in Kayne 1995).” (Fox and Pesetsky (2003) p. 15) We will not examine their proposal for laws of precedence in any more detail, as ultimately I will take them up on their suggestion that the central idea of their theory is not dependent on any particular linearisation algorithm. In fact, I will be using the LCA as the ordering algorithm in my proposal.

### 2.3 Nunes & Uriagereka

Recall Nunes and Uriagereka’s version of the LCA:

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

Clearly, this simplified version of the LCA fails to yield a total ordering on the Lexical Items in many sentences. For example, in a phrase with a complex specifier, there is no asymmetric c-command relation between the elements of the specifier and the sister of the specifier. Nunes and Uriagereka seek to derive such an asymmetric c-command relation using Uriagereka’s notion of Multiple Spellout.

**Multiple Spellout** (Bresnan 1971): Spellout may occur more than once in the course of a derivation.

---

\(^{3}\)It should be noted that Nunes and Uriagereka are not attempting to solve another problem for bare phrase structure and the LCA: the linearisation of simple sisters. If neither sister is complex, and non-branching nodes are disallowed, no asymmetric c-command relation is possible between them. I have no solution to this problem either, and must put it aside as a matter for further research.
Nunes & Uriagereka propose that Spellout linearises according to their LCA (above). After being spelled out, the structure of the constituent is removed from syntax. Only the label, which encodes everything the syntax needs to know to manipulate the spelled-out constituent as a single unit, remains in the derivation. This label, being a terminal node, behaves as a single lexical item. It acts as a “bookmark” to note the location of the spelled-out constituent in the main structure. When PF is generated, the label tells the phonological component where the constituent belongs in the order.

Under this theory, specifiers must be spelled out before merging with the main structure. What is merged is only a label, which appears to the derivation as a terminal node. This simple specifier is in an asymmetric c-command relation with the constituents of its sister.

![Diagram](image)

Figure 1: Lexical Items of a complex specifier have no c-command relationship with Lexical Items of V'
3 Proposal

Both Fox & Pesetsky and Nunes & Uriagereka propose compelling stories of the mechanisms of the Spellout operation. Fox & Pesetsky are able to account for cyclicity effects and order preservation effects, and simplify and reduce the amount of linearisation calculation that must take place to linearise a complex sentence. Nunes & Uriagereka are able to use the LCA, which derives order rather than relying on stipulation. They also account for extraction domain effects such as wh-islands and subject islands: since any Spelled-out constituent is inaccessible to the derivation, it is clearly inaccessible for extraction.

It is tempting to claim they are both correct. The problem with doing so is that Fox & Pesetsky need to be able to access spelled-out constituents to account for object shift, wh-movement, and so one, while Nunes & Uriagereka stipulate that this is impossible. A closer look at the problem reveals that Fox & Pesetsky and Nunes & Uriagereka are looking at different domains. Fox and Pesetsky are interested in the “spine” of the tree and its sisters, while Nunes & Uriagereka are looking at the internal structure of the “satellites”. In technical terms, spine refers to the extended projection line of the semantic head (i.e. the verb) of the tree; informally, this is the main trunk of the tree and the heads thereof. The satellites are essentially the specifiers and adjuncts: anything but the spine. See figure 3 for an illustration of this distinction.

Claim: Both conceptions of Spellout are correct but incomplete. Two separate operations apply: LINEARISE and ATOMISE.
3.1 ATOMISE

ATOMISE is essentially Nunes & Uriagereka’s Spellout. It sends the phonological features of the constituent to the phonological component and in so doing renders the constituent inaccessible to the derivation. ATOMISE applies only when the derivation is “done” with the constituent. The label left behind acts as an atom in the derivation. However, unlike Nunes & Uriagereka’s Spellout, ATOMISE does not linearise the constituent.

3.1.1 LF Spellout

Nunes & Uriagereka claim that after Spellout, “there is literally no syntactic object within [the Spelled-out domain]” (Nunes and Uriagereka (2000) p. 25). If this is true, LF receives no structure whatsoever by the end of the derivation. This is clearly impossible: semantics relies heavily on structure. We are left with two possibilities:

1. The syntactic structure is still there, but simply inaccessible to the syntactic derivation: a closed “suitcase” that can be moved around but which only LF can open.
2. When a constituent is atomised it is indeed removed from the derivation, but it goes both to PF and to LF. In other words, there is also LF Spellout.

It would be ideal to preserve Nunes & Uriagereka’s intuition that the spelled-out constituent is literally gone from the derivation. We would expect much stronger economy effects from the removal of structure from the workspace. This leads us to option two: there exists LF Spellout. Suppose this is true. Then we must ask, does LF Spellout occur simultaneously with PF Spellout? Within this framework, LF Spellout must occur whenever PF spellout occurs; otherwise LF would not receive structure from the derivation. LF-Spellout may also occur separately from atomise, but this can only be true if there are more applications of LF Spellout than atomise.

3.2 LINEARISE

Linearise is Fox & Pesetsky’s Spellout/Linearize, with the ordering algorithm specified as Nunes & Uriagereka’s LCA. Linearise takes a “snapshot” of the derivation so far and applies Nunes & Uriagereka’s LCA, storing the resultant ordered pairs on the Ordering Table. Linearise treats an atomised constituent as a terminal node. This allows satellites to linearise with their sisters.

A linearised constituent is still accessible to the derivation. This solves the problem of accessibility to the spine of the tree. Linearise simply marks the order of Lexical Items.

3.3 Spellout Domains

Like Fox and Pesetsky (2005), I take Spellout Domains to roughly correspond to phases (Chomsky 2001). However, I remain agnostic as to whether there are separate numerations for each phase, or if phases are defined by Spellout Domains.

3.3.1 When spellouts occur

I have no reason to suppose that atomise and Linearise have different domains. However, I do have evidence that Linearise occurs (more or less) as soon as a phase is completed, while atomise is delayed until the next phase is completed. Interestingly, the Phase Impenetrability Condition (Chomsky 2001), being rather ambiguously stated, has been interpreted both as applying as soon as the phase is complete, and as applying after the next phase is complete. I propose that one

4The evocative term “snapshot" comes courtesy of Heather Newell.
spellout operation (ATOMISE) is delayed one phase and the other (LINEARISE) is not delayed.

The evidence that LINEARISE is not delayed comes directly from Fox and Pesetsky (2005). Their theory of linearisation requires that ordering information be recorded as soon as a phase is completed. It is from this that cyclic patterns such as Holmberg effects are derived.

That ATOMISE is delayed is evident from certain Warlpiri data that we will examine in section 6.3 below. The essence of the argument is that multiple elements must escape a phase before it is ATOMISED. Without the delay, there is nowhere for them to move to, as the next phase is not built yet.

3.3.2 Where spellouts occur

I claim that the junctures at which these operations may occur is universal, but that whether they occur or not is parameterised. Ideally, the Spellout Domains should be motivated both theory-internally and by natural language data. I have taken as a starting point the phases suggested by Chomsky (2001): CP, vP and DP. From there I suggest that Cinque (1999)’s heirarchy adds TP as a phase, as the TP region of the sentence is characterised by sentential functional heads, while vP is characterised by the argument structure of the sentence and lexical adverbs, and CP is characterised by propositional attitudes. This distinction seems well-motivated enough, especially given the data from Warlpiri that we will examine in section 6.3 below.

It has also been proposed (Bittner and Hale 1996) that DP can be separated in a parallel fashion: [KP [DP [NP]]] (where KP = case phrase). A simpler [DP [NP]] structure is also perfectly reasonable to assume. The important point is that the Spellout operations have the option of occurring at each of these junctures. We will see evidence in sections 6.2 and 6.3 below that there exists at least one phase edge within the satellites.

A phase is finished when one of these regions of the structure is completed. What signals this completion is not certain. It may be simply that the numeration is completed. If there are not separate numerations for separate phases, the last head of the phase would have to “know” it’s the last. The latter is unlikely in just about any version of bare phrase structure, especially in the TP and CP regions, where the possibility for multiple functional heads is substantial (Cinque 1999). With bare phrase structure, we would expect these heads not to be present if they are semantically and phonologically null.

There may instead be privileged and always-present heads such as v, T, C, and D. This too may be problematic, as many theories of successive cyclic movement
predict a landing site at, for example, the left edge of vP, left of the agent. This would require either another head in the lexical region or a double specifier. The latter is incompatible with Kayne (1994), and the former would predict an optional, unprivileged head to the left of v.

Alternatively, it may be claimed that the merger of a new head that is not of the same “type” (in the sense of Cinque (1999)) as the previous projections triggers the linearisation of the phase. This option relies heavily on a very precise distinction between regions of the spine. It is this last possibility that I will assume to be true, at least for the sake of clear exposition.

It is necessary to stipulate that a shift in head type is not the only motivation for spellout operations. In addition, the completion of the sentence requires that spellout occur. Both spellout operations must be immediately available at this juncture. Atomise must occur, or else parts of the sentence will not be pronounced.\footnote{I am assuming that this is always necessary. It is, of course, logically possible that some languages do not pronounce anything above TP.}

Linearise must be available because clearly many languages have an order to their CP region.\footnote{This is by no means the only way to look at it. Another possibility is to claim that a third type of Spellout exists: an Atomise-like operation that occurs at the completion of an extended projection (CP or DP/KP). This would unfortunately present some difficulties. Linearise would be predicted to not apply to anything above TP. This is clearly not true, so an extended projection Linearise would have to exist as well, bringing the total number of operations to four. So far, the only difference between the two sets of operations is that one is motivated by changing head types and the other by finishing the sentence. Unless stronger evidence arises, Occam’s Razor favours two operations over four. Additionally, movement is possible out of embedded CP: long distance wh-movement occurs. If CP is always Atomised with an EP-Spellout operation, this should not be possible. Perhaps there is something special about the spellout operations that can occur at extended projection edges, but this theory cannot account for it without evoking at least some of the problems listed above. It may be as simple as languages favouring Atomise applying at the completion of CP for reasons of Survival of the Clearest: minimising or eliminating movement between clauses can reduce ambiguity.}

3.4 Example: Building the sentence \textit{The monkey with pizza threw a slice.}

In the interests of sticking to bare phrase structure as much as possible, this declarative English sentence will be assumed to be only a TP.

Step 1. : Merge \textit{threw a slice}
Step 2. Meanwhile, Merge *with pizza* in a separate workspace.

Step 3. Merge *monkey*

At this juncture, the spellout operations may theoretically occur. Since this is English, it makes no difference if they do or not, since the whole DP will be ordered and pronounced, and nothing will move around. For simplicity, then, we will skip this step here.

Step 4. Merge *the*
Step 5. **Linearise** *the monkey with pizza*

<table>
<thead>
<tr>
<th>Ordering Table (DP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>with &lt; pizza</td>
</tr>
<tr>
<td>monkey &lt; with</td>
</tr>
<tr>
<td>monkey &lt; pizza</td>
</tr>
<tr>
<td>the &lt; monkey</td>
</tr>
<tr>
<td>the &lt; with</td>
</tr>
<tr>
<td>the &lt; pizza</td>
</tr>
</tbody>
</table>

Step 6. **ATOMISE** *the monkey with pizza* The phonological features of *the monkey with pizza* are sent to Phonological Component. Let *< the monkey with pizza >* represent the ATOMISED constituent.

Step 7. Merge *< the monkey with pizza >* with *threw a slice*

```plaintext
VP
   / 
  < the monkey with pizza > V' 
      /           \ 
     threw       DP 
        /       
       a slice 
```

Step 8. Merge [+pst] feature

```plaintext
T'
   /     
  +pst  VP
     / 
   < the monkey with pizza > V'
      /           \ 
     threw       DP 
        /       
       a slice 
```
Step 9. Transition from VP region to TP region signaled by Merger of [+pst]: Linearise the VP. (Bolded elements in the table are the new additions)

<table>
<thead>
<tr>
<th>Ordering Table (VP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>with &lt; pizza</td>
</tr>
<tr>
<td>monkey &lt; with</td>
</tr>
<tr>
<td>monkey &lt; pizza</td>
</tr>
<tr>
<td>the &lt; monkey</td>
</tr>
<tr>
<td>the &lt; with</td>
</tr>
<tr>
<td>the &lt; pizza</td>
</tr>
<tr>
<td>a &lt; slice</td>
</tr>
<tr>
<td>threw &lt; a</td>
</tr>
<tr>
<td>threw &lt; slice</td>
</tr>
<tr>
<td>the monkey with pizza &lt; threw</td>
</tr>
<tr>
<td>the monkey with pizza &lt; a</td>
</tr>
<tr>
<td>the monkey with pizza &lt; slice</td>
</tr>
</tbody>
</table>

Step 10. Move the subject to spec-TP

```
TP
   < the monkey with pizza >
      T'
         +pst
            VP
              t
                V'
                  threw
                    DP
                      a slice
```
Step 11. Linearise TP

<table>
<thead>
<tr>
<th>Ordering Table (VP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>with &lt; pizza</td>
</tr>
<tr>
<td>monkey &lt; with</td>
</tr>
<tr>
<td>monkey &lt; pizza</td>
</tr>
<tr>
<td>the &lt; monkey</td>
</tr>
<tr>
<td>the &lt; with</td>
</tr>
<tr>
<td>the &lt; pizza</td>
</tr>
<tr>
<td>a &lt; slice</td>
</tr>
<tr>
<td>threw &lt; a</td>
</tr>
<tr>
<td>threw &lt; slice</td>
</tr>
<tr>
<td>the monkey with pizza &lt; threw</td>
</tr>
<tr>
<td>the monkey with pizza &lt; a</td>
</tr>
<tr>
<td>the monkey with pizza &lt; slice</td>
</tr>
<tr>
<td>+pst &lt; VP = threw a slice</td>
</tr>
<tr>
<td>The monkey with pizza &lt; +pst</td>
</tr>
<tr>
<td>The monkey with pizza &lt; VP = threw a slice</td>
</tr>
</tbody>
</table>

Step 12. Atomise whole sentence

Now we have total ordering on the LIs in the sentence:

The < monkey < with < pizza < +pst < threw < a < slice.

All phonological features are sent to Phonological Component. Notice also that because the monkey with pizza moved before linearise was applied, the fact that it was once below [+pst] is irrelevant. The phonological component was never told that the monkey with pizza was to the right of [+pst].

3.5 Wh-Movement

Consider a pair of wh-questions:

(2)  
  a.  $\text{[CP What}_j \text{ did } [\text{TP } [<\text{the monkey with pizza}>]_i [\text{VP } t_i \text{ throw } t_j ]]}$
  
  b.  $^*\text{[CP What}_j \text{ did } [\text{TP } [<\text{the monkey with } t_j>]_i [\text{VP } t_j \text{ throw a slice}]]}$

(2b) is ungrammatical because it is not possible to extract anything from an atomised domain. (2a) is acceptable since VP was not atomised. Notice also that VP was linearised after what was moved to spec-VP, avoiding the ordering contradiction that might otherwise arise.
4 What do we gain by splitting Spellout?

We have seen that separation of linearise and atomise is possible. Now we will see why it is advantageous. Let us consider what is predicted when each operation acts without the other.

4.1 LINEARISE without ATOMISE

A Spellout operation that allows for accessibility after application makes it possible for such phenomena as Object Shift, Quantifier Movement and wh-movement to occur across linearisation domains. These are the primary applications in Fox & Pesetsky’s 2005 paper on linearisation. Linearise constrains movement across linearisation domains by requiring that any moving elements not change their relative order.

Late Adjunction (Newell 2005, Nissenbaum 2000, Stepanov 2001) may be able to occur after linearise but before atomise. This would allow Late Adjunction to occur after Spellout (of a kind) but still allow atomise to render its domain inaccessible to the derivation. It would probably have to be stipulated that Late Adjoined elements are by default pronounced after the domain to which they are adjoined. A similar situation arises when we consider Lowering (Embick and Noyer 2001, Skinner 2007) to a Spelled-out domain: a linearised domain is accessible but presumably not linearisable. This warrants further research.

4.2 ATOMISE without LINEARISE

Everything in the above section is really just a consequence of Fox & Pesetsky’s conception of Spellout. The following, however, is predicted only by combining Fox & Pesetsky with Nunes & Uriagereka, not by either approach alone.

Normally, we would expect that atomise can only apply to fully linearised constituents; otherwise, elements of that constituent would not be on the Ordering Table. Indeed, Kayne’s Linear Correspondence Axiom is precisely that his linearisation function is a total ordering on the terminal nodes of the sentence. Suppose, though, that atomise applies to an un-linearised domain. Then no information about the order of those constituents will be given to the phonological component. I predict then that scrambling occurs when objects that have not been ordered by linearise are instead assigned a (random) order by the phonological component.\footnote{Or, given that not all grammatical orders in a scrambling language are equal (e.g. Dixon (1972)), pragmatics or even sociolinguistic forces may be involved in the choices between grammatical orders.}

7
5 Scrambling Typology

Because it is possible to separately linearise and atomise different parts of the sentence, combinations of these two operations in various sections should derive many types of Scrambling. Consider the following model:

Suppose there are five basic Spellout Domains:

- **S** - The whole sentence
- **T** - TP
- **V** - vP
- **D** - Whole satellites (e.g.: DP)
- **N** - Subpart of satellites (e.g.: NP)

Suppose further that each of these domains can be linearised or not, and atomised or not. Call these combinations Scrambling Types, as follows:

\[
S[\pm L\pm A] \quad T[\pm L\pm A] \quad V[\pm L\pm A] \quad D[\pm L\pm A] \quad N[\pm L\pm A]
\]

For example, Scrambling Type

\[
S[-L+A] \quad T[\pm L\pm A] \quad V[+L-A] \quad D[-L+A] \quad N[\pm L\pm A]
\]

has un-linearised but atomised satellites, linearised but un-atomised vP and un-linearised but atomised whole sentence. The other Spellout Domains are undetermined.

Now, in this schema, there are 1024 logically possible Scrambling Types:

\[
(2^2)^5 = 1024
\]

However, three things can make a scrambling type impossible:

1. The final sentence is not atomised and therefore PF never receives its phonological features. This eliminates half of the possibilities.

2. Linearisation problems occur when satellites (D) are un-atomised but the spine is linearised. Linearise requires that the LCA be used, but un-atomised complex satellites make that impossible.

3. Not Scrambling: everything is linearised before being atomised.

It must also be considered that true randomness is not found outside of quantum mechanics, so this “randomness” must be governed by something. The important thing, though, is that linear order is not linked to the syntax or semantics.
5.1 A Sampling of Types


Figure 4: (Framed constituents are ATOMISED.)

→ Scrambling within satellites only
→ Example word orders:

- C T ABG V DEF
- C T AGB V EFD
- C T GBA V FED
- C T ABG V DFE

Figure 5: (Framed constituents are ATOMISED.)

→ Scrambling of LINEARISED, ATOMISED satellites within TP; C-initial
→ Example Word orders:

- $C \; [V \; [ABG] \; T \; [DEF] \; ]$
- $C \; [ \; [ABG] \; [DEF] \; T \; V \; ]$
- $C \; [ \; [DEF] \; V \; [ABG] \; T \; ]$
- $C \; [T \; V \; [DEF] \; [ABG] \; ]$
3. \textbf{S}[-L\text{+}\text{A}] \textbf{T}[-L\text{−}\text{A}] \textbf{V}[-L\text{−}\text{A}] \textbf{D}[-L\text{−}\text{A}] \textbf{N}[-L\text{−}\text{A}]

\begin{center}
\begin{tikzpicture}
  \node (CP) {CP};
  \node (C) [below of=CP] {C};
  \node (TP) [right of=C] {TP};
  \node (T) [below of=TP] {T};
  \node (vP) [below of=T] {vP};
  \node (DP) [above right of=vP] {V'};
  \node (A) [left of=DP] {A B G};
  \node (V) [right of=DP] {D E F};
  \node (D) [below of=DP] {D P};
  \node (E) [above of=V] {E};
  \node (F) [right of=V] {F};
  \node (G) [left of=V] {G};

  \draw (CP) -- (C);
  \draw (C) -- (TP);
  \draw (TP) -- (T);
  \draw (T) -- (vP);
  \draw (vP) -- (DP);
  \draw (DP) -- (A);
  \draw (DP) -- (V);
  \draw (V) -- (D);
  \draw (V) -- (E);
  \draw (V) -- (F);
  \draw (V) -- (G);
\end{tikzpicture}
\end{center}

\textbf{Figure 6:} (Framed constituents are ATOMISED.)

$\rightarrow$ Totally free word order
$\rightarrow$ Example Word Orders:

- A G C V T D F B E
- V F G E T A C B D
- A B C D E F G V T

4. \textbf{S}[-L\text{+}\text{A}] \textbf{T}[-L\text{−}\text{A}] \textbf{V}[+L\text{−}\text{A}] \textbf{D} [\pm L\text{+}\text{A}] \textbf{N}[\pm L\pm A]

$\rightarrow$ Set relative order for a lower SD (V), but these \textsc{linearised} LIIs would be able to Scramble with constituents in the rest of the sentence.

$\rightarrow$ Resultant sentences would have same order for \textsc{vP}-satellites, but everything merged above \textsc{vP} can appear in any order, even intermingled with \textsc{vP} elements, whether they’ve moved out of \textsc{vP} or not.

Type (4) is an interesting prediction this approach makes. Let us examine it a little more closely in figure below.
X and Y have moved out of vP and even maintained their relative order, but this will not matter, as LINEARISE does not apply again. LINEARISATION of vP determines the order $X < Y < Z$. Because the CP is ATOMISED without LINEARISATION, A, B, and C have no place on the Ordering Table. Therefore $A, B, C, X, Y, Z$ can order freely as long as $X < Y < Z$. 

<table>
<thead>
<tr>
<th>Ordering Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X &lt; Y$</td>
</tr>
<tr>
<td>$Y &lt; Z$</td>
</tr>
<tr>
<td>$X &lt; Z$</td>
</tr>
</tbody>
</table>
→ Example Word Orders:

- B X Y C A Z
- A B C X Y Z
- X B Y A Z C

A scrambling type like this would be difficult to explain in any other framework, and would give good evidence that this is the right way to approach scrambling.

6 Scrambling Data – Some Examples

6.1 Dyirbal

Dyirbal is a non-configurational language. This means it has free word order, syntactically discontinuous expressions, and extensive use of null anaphora (Hale 1983). Hale considered these three properties aspects of a single parameter. If Dyirbal is of type $S[-L+A]$ $T[-L-A]$ $V[-L-A]$ $D[-L-A]$ $N[-L-A]$ (type 3 above), the first two properties are easily explained as being aspects of a single parameter. Consider the following data: (Dixon 1972)$^{8,9,10}$

(3) a. bayi wangal bangul yarangu bulganu banggun dugumbiru buran woman-ERG see-PRES/PST

‘The woman saw the big man’s boomerang’

b. bayi yarangu dugumbiru buran wangal banggun bangul bulganu

the-GEN woman-ERG see-PRES/PST boomerang-NOM the-ERG the-GEN big-GEN

---

$^8$Orthography is slightly modified to use ASCII characters. ng = velar nasal, r = “semi-retroflex continuant”, d = “lamino-palatal/alveolar” (Dixon 1972 p. 2).

$^9$the in the glosses is my approximation of a more complex, but not here relevent, noun-marker system.

$^{10}$Abbreviations are as follows: NOM = nominative case GEN = genitive case ERG = ergative case PRES/PAST = present or past tense.
The **bold**, *italics* and underlining are to clarify which elements have the same case. This second order is truly amazing, with nominal markers separated from their nouns and adjectives separated from that which they modify. Nor are these the only possible orders: Dixon states clearly that all word-orders are grammatical. Example (3b) was chosen because this order was “made up” by Dixon to illustrate dramatic scrambling. He writes:

A well-known linguist took exception to this, categorically denied that freedom of word-order of this magnitude was possible in any language, and accused the writer of exaggerating. (321) [3b] was put to informants at the next opportunity, and they castigated the writer for asking a trivial and unnecessary question – “you know that’s alright!” (Dixon 1972 p. 107-8)

Aside from the free word order, the thing to notice is that the noun markers can be separated from the nouns they modify. I propose that this occurs when satellites are not **atomised** before being merged to the tree. This is possible if and only if the constituent of which the noun and noun marker are a part is **atomised** without being **linearised**; otherwise the un-**atomised** constituent would cause problems for the LCA.

This sentence has eight words. This means there are 8! = 40 320 possible word-orders. If these word-orders were produced by optional movement, there would have to be 40 320 different optional movement combinations.

Non-configurational languages are often assumed to have a flat structure for at least part of the sentence (Hale 1983). The advantage of my approach is that the absolute randomness of order can be achieved without assuming very different structures for configurational and non-configurational languages.

### 6.2 Tagalog

I propose that Tagalog is of type $S[+L+A]$ $T[-L+A]$ $V[-L-A]$ $D[+L+A]$ $N[-L+A]$ : i.e. its **atomised** satellites scramble within TP and word order is free inside NP. Consider the following data\(^{11}\)

\(^{11}\)Data from Kroeger (1993)

\(^{12}\)Abbreviations are as follows:

NOM = nominative case
GEN = genitive case
DAT = dative case
LK = linker

---

22
We can see from these word-orders that satellites are freely ordered within TP, but the verb is in the initial position. This can be explained if the sentence is built thus:

1. DPs are linearised and atomised before being Merged.
2. V moves from its merged position to the first head in the CP range
3. TP is atomised without being linearised
4. CP is linearised and then atomised

When an adjective is added, more is revealed. The case marker is consistently in first position within the DP, but the adjective and noun can be in either order.\(^{13}\)

These data are explained if NP, in parallel with vP, is a phase. In Tagalog, NP is atomised without being linearised, resulting in scrambling within NP. DP, though, is linearised and atomised, deriving a consistent linear location for the case marker (which I am assuming is D, head of DP).\(^{14}\)

The above data combine to give twelve possible orders for the following sentence:

\(^{13}\)Data from Raphael Marcado, p.c.
\(^{14}\)Throughout this paper, it is perhaps more theoretically consistent to include KP in the structure, especially as most of the “determiners” are really case markers. I have chosen to keep the satellites to only two phases because I have no evidence for more than two. For now, DPs and KPs can be thought of as equivalent.
The final question is whether Tagalog allows discontinuous expressions. The evidence I have gathered so far indicates that although discontinuous expressions are possible, though they are not possible in the same way as they are in Dyirbal. Consider the following contrast:\(^{15}\):

9. a. nagbigay ng libro sa babae ang lalaki na malaki
   gave GEN book DAT woman NOM man LK big
   ‘The man gave the woman the big book.’ OR ‘The big man gave the woman the book’

   b. nagbigay ng libro sa babae ang lalaki-ng malaki]
   gave GEN book DAT woman NOM man-LK big
   ‘The big man gave the woman the book’

The only difference between these two sentences is the form of the linker. \textit{Na} is the full form, and \textit{ng} is the cliticised form. I suspect that \textit{na malaki ‘LK book’} is in fact some kind of relative clause, approximately equivalent to the English \textit{and it was big}. This would explain its ability to stand on its own. Further evidence comes from the behaviour of ordinary relative clauses:

\(^{15}\)Data from Raphael Marcado, p.c.
(10) a. nagbigay ng libro sa babae ang lalaki [na binili niya sa
gave GEN book DAT woman NOM man [LK bought TT 3sg OBL
maynila]
Manila]
‘The man gave a book that he had bought in Manila to the woman.’
b. ? nagbigay ng libro-ng binili niya sa maynila] sa babae
gave GEN book-LK [bought TT 3sg OBL Manila] DAT woman
ang lalaki.
NOM man

When the relative clause is stranded and acceptable, it is linked by a full-form
linker na. The sentence is awkward when the relative clause is sister to the head
it modifies. In that case, it has a clitic linker ng. I conclude that relative clauses
are generally merged separately, and not as part of a DP. From this I predict that
relative clauses scramble with the rest of the satellites, as long as the linker is in its
full form. More data will confirm or refute this claim.

Similarly, I propose that the stranded adjectival modifier is a relative clause, and
therefore patterns in the same way I predict ordinary relative clauses to pattern.
The reason the adjective can appear in the DP with the noun it modifies and with
the clitic linker is that adjectives can also appear as ordinary modifiers, not only as
relative clauses.

6.3 Warlpiri

It is common knowledge that Warlpiri has discontinuous expressions. In the examples
below, adjectives are separated from the nouns they modify.

(11) a. Maliki wiri-ngki ø-ji yalku-rnu
dog big-ERG PERF-1-OBJ bite-PAST
‘The/a big dog bit me’
b. Maliki-rli ø-ji yarliku-rnu wiri-ngki
dog-ERG ø-ji PERF-1-OBJ bite-PAST big-ERG

(12) Nyangu jana rdaku walyangka wita-wita manu wiri-wiri
saw 3-PL-DAT hole ground-LOC small-small and big-big

\[^{16}\text{Data from Hale 1983. New abbreviations for Warlpiri: PERF = perfective; NPST = non-past; 1,2,3 = 1^{st}, 2^{nd}, 3^{rd} person; SG = singular; PL = plural; S = subject; LOC = locative; KN = known}\]
‘He saw their big and little holes in the ground.’ \(^{17}\)

Despite the apparent discontinuity of DPs, I claim for two reasons that satellites are \textit{atomised}.

First, in some cases, DPs must appear continuously. If there is only one case marker, the adjective cannot be separated from the noun. Moreover, the modifier always follows the modified expression and the case-marker must appear at the end of the expression, cliticised to the modifier.

\begin{enumerate}
\item \textit{warna maru-ngku}  
\textit{snake black-ERG}  
\textit{‘The/a big snake’}
\item \textit{warna-ngku maru-ngku}  
\textit{snake-ERG black-ERG}
\item * \textit{warna-ERG maru}  
\textit{snake-ERG black}
\end{enumerate}

Second, both Hale and Laughren frequently translate discontinuous expressions as secondary predication. An example is the second translation offered for (12) above: ‘He saw their holes in the ground, small ones and big ones.’. Further examples can be seen in (14) and (15) below:

\begin{enumerate}
\item \textit{Maliki-rli ka marlu wajilipi-nyi mata-ngku}  
\textit{dog-ERG PRES kangaroo chase-NPST tired-ERG}  
\textit{‘The dog, tired, is chasing the kangaroo’}
\item \textit{Mata-ngku ka marlu wajilipi-nyi maliki-rli}  
\textit{tired-ERG PRES kangaroo chase-NPST dog-ERG}
\end{enumerate}

\begin{enumerate}
\item \textit{Maliki-rli ka marlu wajilipi-nyi mata}  
\textit{dog-ERG PRES kangaroo chase-NPST tired-ERG}  
\textit{‘The dog is chasing the kangaroo (and the latter is) tired’}
\item \textit{Mata ka wajilipi-nyi maliki-rli marlu}  
\textit{tired PRES chase-NPST dog-ERG kangaroo}
\end{enumerate}

The predication is not subject to binding as it is in English (Hale 1994), as is evident in the awkward translation of (15). This is consistent with true (non-movement) scrambling.

\(^{17}\)Data from Mary Laughren 2007 p.c.
Putting these two observations together, one can conclude that singly-case-marked and doubly-case-marked expressions are distinct entities. I suggest that doubly-case-marked expressions such as *warna-ngku maru-ngku* ‘black snake’ are in fact secondary predicates and are merged separately. Singly-case-marked expressions such as *warna maru-ngku* ‘black snake’, on the other hand, are linearised, atomised DPs. The phrase marker below is an example of how this might work, within the Kaynian framework. The boxed NP is linearised and atomised before being moved to spec-DP. This is evidence of NP as a phase.

\[
\begin{array}{c}
\text{DP} \\
\text{warna maru} \\
\text{D'} \\
\text{ngku} \\
\text{NP} \\
\text{warna maru}
\end{array}
\]

This DP is then linearised and atomised before being Merged to the main tree, yielding both the set order of elements within the DP and the continuousness of the expression.

As for the spine of a Warlpiri sentence, the evidence indicates that elements are freely ordered in TP but the elements in CP are strictly ordered. Consider the following data: 18

(16)  
*Karinganta wangumarnanypa-lku waja-npa nyuntu-ju*  
I-assert orphan-now I.say-2-sg you  
‘It’s that you are now an orphan’ (In reply to the question, ”Why are you two crying?”)

(17)  
*Pangurnu-ju nyarrpara-wiyi ka-nkulu marda-ni?*  
shovel-kn where-before PRES-you-S hold-NPST  
‘Where have you got a shovel?’ (In reply to ‘Let’s dig with a shovel.’)

The elements in **bold** are pre-auxiliary elements. *Karinganta* is an evidential Speech Act particle. Contrary to traditional belief, more than one element may appear before AUX. 19 (16) shows that evidentials are possible before AUX, and (17)

---

18Data from Simpson (in press)  
19It may be an oversimplification to assume that the auxiliary is a complex head that has moved to the first head position in the CP range. Laughren, for example, splits the AUX into an aspect
shows that both a focus and topic can appear in pre-aux position: in other words, elements of a complex CP region (Simpson in press). Moreover, these elements are strictly ordered.

While the evidential *karinganta* is presumably merged in the CP region, and the pre-aux verb is likely picked up by the same head movement that forms the auxiliary, it is probable that the *wh*-phrase *nyarrpara-wiyi* ‘where-before’ and the noun *wangumarnanypa-lku* ‘orphan-now’ come from the theta structure of the VP. Warlpiri word order is free below AUX, and AUX appears to be on the right periphery of the CP region. It should therefore be assumed that VP is neither linearised nor atomised, but that TP is atomised. TP (containing VP) must not be atomised until after the topic and/or focus is moved out. In order for these elements to have a landing-place, at least one, and sometimes, two, full phrases must be built before TP can be atomised. It is Warlpiri that makes it clear that atomise is delayed by one phase. Even if instead atomise applied after the next phrase was built (i.e. the phrase that contains TP), only one satellite would be able to escape TP before it was atomised. If instead atomise is delayed until the completion of the next phrase – in this case, CP – then as many satellites as necessary can escape, but what remains in TP can still be scrambled.

My conclusion is that Warlpiri atomises TP without linearisation, and both linearises and atomises CP. When we compare this result with the above analysis of Tagalog, the surprising conclusion is that Warlpiri word order is stricter than Tagalog, as the only difference is that Tagalog is $N[-L+A]$ and Warlpiri is $N[+L+A]$. That is, Warlpiri linearises its NPs but Tagalog does not.

### 6.4 Tohono O’odham

Tohono O’odham (formerly known as Papago) is an Uto-Aztecan language of southern Arizona and northern Mexico.\(^{20}\) It has about 20 000 speakers. It is a discourse configurational language. I propose it is of scrambling type

$$S[+L+A] \ T[-L+A] \ V[-L-A] \ D[+L+A] \ N[\pm L\pm A].\(^{21}\)$$

This makes it essentially identical to Warlpiri and Tagalog. I include it for three reasons. First, more data indicates that there is indeed something regular and universal about the apparent patterns I have identified. Second, its striking similarity marker and pronominal clitics. This debate is beyond the scope of this paper.

\(^{20}\) All data and information on this language are from Miyashita et al. (2003)

\(^{21}\) The $N[\pm L\pm A]$ part is temporary, I hope. It is such because I have no data of DPs any more complicated than $[\text{det} + N]$. 

28
with distant languages is interesting in itself. And third, O’odham has a most un-
expected property: despite its near total freedom of word order, it is not a very
morphologically rich language. In particular, the third person singular marker is
null. This leads to ambiguity in very simple sentences.

Like Tagalog and Warlpiri, O’odham has a privileged “second” position, before
which a satellite may appear and after which word order is free. The examples below
have a third person subject and object, and as such are ambiguous.\(^{22}\)

(18) a. Huan ’o g Husi ka:
   John 3sub-imp det Joe hear
   ‘John is hearing Joe’ OR ’Joe is hearing John’

b. Huan ’o ka: g Husi

c. Husi ’o g Huan ka:

d. Ka: ’o g Huan g Husi

e. Ka: ’o g husi g Huan

The same two readings are also available when one of the arguments is not overt:

(19) a. Ka: ’o g o’odham
   hear 3sub-imp det man
   ‘The man is hearing him/her/it’ OR ’He/she/it is hearing the man’

b. O’odham ’o ka:

The initial position is also used for interrogative pronouns. The same ambiguity
can arise:

(20) Do: ’o ŕeïd g Mali:ya?
   who 3sub-imp see Mary
   ‘Who is seeing Mary?’ OR ’Who is Mary seeing?’

Like Warlpiri and Tagalog, I believe the auxilliary is a complex head of which
TP is the complement. In wh-questions, the wh-word moves into spec-CP. In other
sentences, any XP may move into spec-CP.

\(^{22}\)Abbreviations: 3 = third person; sub = subject; imp = imperfect; det = determiner. The
determiner g carries no information.
7 Conclusions

7.1 Summary

Syntactic Spellout is at least two separate operations: LINEARISE defines the linear order of terminal nodes; ATOMISE sends phonological features to the phonological component, rendering the ATOMISED domain inaccessible to the syntactic derivation.

If LINEARISE applies without ATOMISE, phenomena such as Holmberg’s Generalisation, wh-movement, and other such cyclic movements are able to occur. LINEARISE allows movement out of a LINEARISED constituent, while constraining final order.

If ATOMISE applies without LINEARISE, scrambling occurs within the ATOMISED constituent. The ATOMISATION without LINEARISATION of different constituents accounts for and predicts different scrambling types.

7.2 Further Research

The preliminary nature of this research means there are numerous unanswered questions; I will list only a few here.

So far, all the evidence indicates that ATOMISATION and LINEARISATION domains are identical. If this is true, further evidence from scrambling and other cyclicity effects will continue to support the hypothesis. If they are in fact different domains, further data should indicate that.

LF-Spellout is a vital part of any theory of Spellout that claims that Spellout literally removes spelled-out constituents from the derivation. Either the constituent is spelled out simultaneously to LF or the constituent is not truly removed from the derivation. There are theories that claim LF- and PF-Spellouts occur separately (e.g.: Marušić (2005)) and that they occur together. These matters deserve further research, in particular to see how LF-Spellout domains might interact with applications of ATOMISE. For example, it is possible that a single parameter controls whether ATOMISE occurs and whether LF-Spellout occurs at a particular juncture. Languages that go for several phases without ATOMISING may then also go for several phases without spelling out to LF. Only further data can shed light on this question.

Lowering and Late Adjunction are interesting research directions, as both may occur post-Spellout. As LINEARISE allows accessibility to its domain, this interpretation of Spellout may be of interest. However, the question of ordering the Lowered or Late-Adjoined element is a serious issue, and may even lead to a modification of this entire theory. Further research is necessary to see if this is a fruitful avenue of exploration. Also in this domain of research is how these operations might interact with Distributed Morphology (Cf: Skinner (2007)).
How is it possible to \textsc{linearise} simple sisters? Possible solutions are that Bare Phrase Structure is too bare, or that the LCA is not the ordering algorithm \textsc{linearise} uses.

Finally, my proposal predicts many scrambling types not explored here. Data from additional scrambling languages can help test and refine this model. If no more scrambling types are to be found in natural language, there is certainly something wrong with this theory. If further scrambling data conforms to predictions by this model, the model may in fact be valuable. Languages of interest include Classical Greek, Latin, and Hungarian.

7.3 Theoretical Consequences

If my proposal is correct, the originally simply-shaped model of syntax is yet less simple. Not only are there multiple branches from the derivation to PF, but there are (at least) two different types of branches to PF: one for \textsc{linearise} (depicted as boxes) and one for \textsc{atomise} (depicted as arrows). There are also branches going the other direction: to the conceptual-intentional component of language.

![Three Models of Spellout](image)

Figure 8: Three Models of Spellout

Different types of information are given to the phonological component at different times. It seems to me that the recipient of the information \textsc{linearise} and \textsc{atomise} generate is better thought of as the phonological component rather than the phonological form (PF), given how much processing the phonological component is assumed to be doing in order to generate the Phonological Form. If I am correct, the
phonological component is processing not only phonological features, but also reinsertion of previously ATOMISED constituents in the linear order, interpretation of the Ordering Tables, and order-assignment to un-LINEARISED domains. The distinction between the syntax as a process and PF as a form may continue to weaken, especially with further research into how morphology might fit into this model. Whether this will prove beneficial or detrimental to the pursuit of a minimalist, computational system of language is only one of many exciting questions to explore.

References


Bresnan, Joan. 1971. Contraction and the transformational cycle in English. Ms., MIT, Cambridge, MA.


Fox, Danny, and David Pesetsky. 2003. Cyclic linearization and the typology of movement. online.


