Multiple Spell-out[1]

Juan Uriagereka
University of Maryland

A presently unmotivated assumption within the Minimalist program is that the rule of Spell-out applies only once. This paper argues that such a residue of the level of S-structure is undesirable, and liberalizing Spell-out to apply as many times as it needs to, up to economy, has several pleasant features, as well as various intriguing consequences which are worth pursuing. The model that emerges is radically derivational, so much so that in the end the issue is raised of whether the PF and LF components organize themselves into traditional levels of representation.

I Deducing the Base-step of the Linear Correspondence Axiom

One of the central desiderata of the Minimalist Program is reducing all substantive principles to interface (or bare output) conditions, and all formal principles to economy conditions. As a result, much energy has been devoted to rethinking constraints and phenomena which do not appear to conform to this desideratum as more elementary ones, in the process focusing the data base. This paper is an attempt to reduce a version of Kayne's (1994) Linear Correspondence Axiom (LCA) to more minimalist bases.

Chomsky (1995:chapter 4) already limits the place for the LCA in the grammar. Kayne's version of the axiom is a formal condition on the shape of phrase markers (PM). Chomsky's (both for empirical and conceptual reasons) is a condition that operates at the point of Spell-out, and because of PF demands. The intuition for Kayne is that a non-linearized PM is ill-formed, in and of itself, whereas for Chomsky such an object is ill-formed only at PF--hence the demand to linearize it upon branching to this component. Chomsky's version is more minimalist in that the linearization condition is taken to follow from bare output conditions.

The axiom has both a formal and a substantive character. The formal part demands the linearization of a complex object that has been assembled by the Merge operation, which produces mere associations among terms. A visual image to keep in mind is a mobile by Alexander Calder: the hanging pieces relate to each other in a fixed way, but are not linearly ordered with respect to one another; one way to linearize this object is to freeze it in time, for instance through a photograph. The substantive part of Kayne's axiom does the job of the photograph: it tells us how to map the unordered set of terms into a sequence of PF slots. Now, even if Chomsky's reasoning helps us deduce the formal part of the axiom (assuming PF demands linearization), the question still remains of <u>how</u> the mapping works--how we 'take the photograph'.

Kayne is quite explicit about this. Unfairly, I will attempt to adapt his ideas to Chomsky's 'bare phrase structure', which is again more minimalist than Kayne's version of this structure. (Chomsky avoids committing to a particular definition.) We may restate Kayne's axiom as in (1):

(1) <u>Linear Correspondence Axiom</u>
a. Base Step: If @ commands &, then @ precedes &.
b. Induction Step: If $ precedes & and $ dominates @, then @ precedes &.

I will discuss each of the steps in (1) in turn, with an eye on deducing their substantive character from either bare output or economy considerations. Consider then why it should be that command is a sufficient condition for precedence. It is best to ask this question with a particular formal object in mind. I will call this object a <u>command unit</u>, for the simple reason that it emerges in a derivation through the monotonic application of Merge. That is, if we choose to merge elements to an already merged PM, then we obtain a command unit, as in (2a). In contrast, (2b) is not a command unit, since it implies the non-monotonic application of Merge:

```
(2)  a.  Command unit:                       b.  Not a command unit:
         {@,{$,{@,{@,{&....}}}}}                {@,{{$,{$,{%...}}},{@,{@,{&...}}}}}
            ^                                                ^
         $<-l->{@,{@,{&...}}}                   {$,{$,{%...}}}<-l->{@,{@,{&...}}}
                  ^                                  ^               ^
               @<-l->{&...}                      $<-l->{%...}       @<-l->{&...}
     Monotonic application of Merge to same   Non-monotonic application of Merge,
     object to two separately assembled objects
```

Considering the matter for command units, the real question is why, among the possible linearizations in (3), (3d) is chosen:

```
(3) a.  {@,...}        b. {@,...}      c. {@,...}      d. {@,...}      e. {@,...}       f. {@,...}
         /  \              /   \           /  \            /  \             /|                |\
     {@...}  \          {@...}   \        /  {@...}       /  {@...}     {@...|}           {@|...}
        /  \   \          /  \      \     /      /  \     /      /  \    /   |\              /| \
   {&...}  @    $       @ {&...} $     $ {&...} @      $   @ {&...} {&...} $ @            @ $ {&...}
```

Keep in mind that, although branches cross in the diagrams in (3e) and (3f), this is an artifact of the two-dimensional representation given here. It is perhaps best to imagine all situations in (3) as mere states of a hierarchical Merge function, which creates several 'merge waves': one where {@...} and $ interact, one where @ and {&...} interact, and so on. These 'waves' may be thought of as 'collapsing' into given linearizations. Then the task is to figure out why the <$, @, {&...}> 'collapse' is the one chosen by Human language, and not the other, apparently reasonable ones.

We may ask the question from the point of view of what syntactic relations are relevant to the terminals of the structures in (3). If understood as 'merge waves', the only relation that exists in these objects is 'I have merged with you.' We obviously can produce an order in terms of this relation, which essentially keeps track of who has merged with who when. This is, essentially, the insight behind Sam Epstein's (forthcoming) interpretation of command, which has the effect of ordering the terminal objects in (3) as follows: <$, @, {&...}>. If because of PF reason the system demands that merge waves collapse into flat objects, it is not unreasonable to expect that it does so in terms of a previously existing relation. Of course, 'not unreasonable' does not mean 'right'; nonetheless, within minimalist assumptions, we expect this sort of harmony in the system. Basically, the idea is to piggy-back on an already existing order to determine PF order.

However, we have not achieved the desired results yet. To see this, imagine a group of people trapped inside a building, and having access to a window that only allows a one-at-a-time exit. These people may order themselves according to some previously existing relation (e.g. age), so as not to dispute on a new order. But having found an order does not mean having decided how to come out; does the youngest come first or last? Why not start in the middle? And so on. Likewise, a decision has to be made with regards to the <$, @, {&...}> order. Exactly how do we map it to the PF order? Obviously, many answers come to mind.

Here is where the minimalist thinking in terms of optimization can be useful. The question may not just be how to map the sequence <$, @, {&...}> to a PF order, but actually how to do it <u>optimally</u>. Is there a best solution to the problem?

The hope is that mapping the <$, @, {&...}> command order to the <$, @, {&...}> PF order in (3d) is the best solution there is. The good news is that, on reasonable assumptions about the information cost of alternative mappings, this is indeed the case. In particular, note that the function mapping a <$, @, {&...}> order to a <$, @, {&...}> order is mere identity. The reader can try as an exercise stating a precise function to map, instead, the <$, @, {&...}> command order to any of the other PF orders in (3); whatever this function is, it simply will be more complex than identity. If information cost enters the computation of complexity among different ways of harmonizing the command relations with corresponding PF relations, then (3d) wins, as desired.

But there are some bad news as well. The visual picture in (3) is misleading in that it assumes the convention of mapping left-to-right script to before-and-after speech. Suppose this book were written in Hebrew. Why couldn't we then interpret (3a)--the reverse image of (3d)--as the right solution? Or to put it more clearly: why have we chosen to map the command relation to precedence? Why not map it to the, as far as I can see, equally reasonable following relation? Plainly, the way it has been stated, the problem has not one, but in fact two optimal solutions.

I think there are only three sorts of things one can say at this juncture. One is to blame the choice of (3d) over (3a) (now taking left-to-right to mean before-and-after) on some deep principle of this universe. It would have to be as deep as the second law of thermodynamics, though--and I do not mean this just as a joke. That is, the would-be principle would have to be as profound as whatever explains the forward movement of time, which standard physical equations do not explain. I haven't been able to convince anyone, including myself, that such a principle exists.

The second thing one can try to say, assuming that (3a) and (3d) are equally optimal soltions, is that there was an adaptive edge that (3d) gave the human species. To me, at least, this seems an even fancier story to tell than the thermodynamics one. A species that had chosen (3a) instead of (3d) might have equally well have stumbled onto, or somehow adapted into, a parser and an acquisition device for the relevant structures. (The reason for my skepticism might just be ignorance about parsing and acquisition.)

The third alternative is to shrugh one's shoulders. So what if (3a) and (3d) are equally economic? You have two equally valid solutions, so randomly pick one that does the work. This view of the world would certainly be very consistent with the perspective of Punctuated Equilibrium in biology, of the sort Stephen Gould and his associates advocate for. More importantly for us here, such a view is perfectly consistent with the Minimalist program, which contrary to popular belief does not seek the best solution to optimality problems, but just a best solution.

If I am on track, (3d) is correctly chosen as the actual PF ordering that the system employs; that is, we do not need to state (1a) as an axiom--it is true, but derived. In a nutshell, command maps to a PF linearization convention in simple command units (those which (1a) is designed to target) because this state of affairs is optimal. This decision is not reached in a vacuum; it is true only within a specific frame of reference, and removing the subtle assumptions behind the reasoning simply destroys it. One such assumption is evidently minimalist: choosing among the derivational alternatives in (3) through an optimization process. (These are indeed derivational alternatives, given Chomsky's suggestion that timing is a matter of PF.) A less obvious assumption, but still within the minimalist spirit, is that information considerations should be the basis for the optimization procedure. What the derivation of the LCA does not do is tell us whether the PF linearization convention is 'precede' or 'follow'. There is, however, no obvious reason why the linguistic system itself should decide on this matter; in any case, even if there were functional or even 'deep' reasons for this choice, the rest of the reasoning would remain intact.

II Deducing the Induction-step of the Linear Correspondence Axiom

In the previous section we have met the demands of the Minimalist program by reducing part of the LCA to more justifiable conditions. Naturally, we should ask ourselves whether we can deduce the whole LCA this way. In short, I know of no deduction, of the sort we have seen above, given standard assumptions about the model at large.

Nonetheless, there is an intriguing way of going about the general deduction of the LCA, if we go back to a model that was popular in the seventies.[2] For reasons that become apparent shortly, I refer to it as a dynamically split model. The origins of this outlook are discussions on successive cyclicity, and whether this condition affects interpretation. Much of the debate was abandoned the moment a single level, S-structure, was postulated as the interface to the interpretive components. But before that, the question arose as to whether access to these components was done in successive bursts of derivational cascades--as (derivational) time goes by. Now that, with D-structure, we have abandoned S-structure, the question is alive again: What would it take for the system to access the split paths of interpretation in a dynamic way? What would it mean to do so?

I want to demonstrate that the simplest assumption is one in which the dynamically split access to interpretation is possible, and furthermore that by doing so, the simplified system satisfies the Induction-step of the LCA in a theorematic way, thus allowing us to entirely dispense with the statement as an axiom.

One way of framing the issue is asking how many times the rule of Spell-out should apply. If only once, we must have a single access to PF and LF, precisely at that point. On the other hand, accessing PF and LF in successive derivational cascades entails a multiple application of Spell-out. Surely, economy considerations alone favor a single application of Spell-out, assuming that a derivation involving less computational steps is less costly.[3] But are there circumstances in which a derivation is forced to spell-out different chunks of structure in different steps?

One such instance might arise when a derivation involves more than one command unit. As we saw, command units emerge naturally as the derivational process unfolds, and they are trivially linearized through the base-step of the LCA, now deduced. So what if only those trivially linearizable chunks of structure (such as (2a)) are in fact linearized? That is, what if, instead of complicating the LCA as in (1b), when we face a complex structure of the sort in (2b) we simply do not linearize it, leading to a derivational crash? Only two results are then logically possible: either structures of the sort in (2b) do not exist, or we linearize them in various steps, each of which involves only command units. The first possibility is factually wrong, hence we conclude that multiple Spell-out is a reasonable alternative.

Before we explore whether multiple Spell-out is also an empirically desirable alternative, let us consider its mechanics. First, we must bear in mind that command units are singly spelled-out, since that is the most economic alternative. But the real stuff takes place beyond command units. By assumption, we have no way of linearizing them, so we must do the job prior to their merger, when the component parts are still command units. Then we must come up with a procedure to merge a structure which has already been spelled-out, or else we would not be able to assemble the final unified object in each case. I propose the following.

Spell-out separates phonetic from categorial/semantic features, yielding structures which are interpretable by the PF and LF components. Literally, what remains after the split is no longer a PM, and is not accessible by any further transformational rules. If we are to use an analogy, the spelled-out PM is like a lexical compound, whose syntactic terms are interpretable but are not accessible to movement, ellipsis, and so forth. Needless to say, the reason why the compound is 'frozen' and the spelled-out PM is 'frozen' are completely different--but the formal effect is alike.

There is one other sense in which the spelled-out PM is like a compound: it must behave as a word, so that it can merge further up to the rest of the structure; this means the PM must keep its label after Spell-out. Technically, if we have a PM {@,{L,K}} to linearize through Spell-out, what we must obtain is {@,<L,K>}, which is mathematically equivalent to {@,{{L},{L,K}}}.[4] It is easy and curious to show that this object is not a syntactic object. If this is true, the reason why it behaves as a sort of 'frozen' compound is direct, and we need not stipulate the result beyond what we already have: the linearization procedure of Spell-out, which is motivated through bare output conditions on PF.

To see how we reach this conclusion, we must take very seriously the notion of syntactic object introduced in Chomsky 1995:chapter 4, which must have one of the two formats in (4):

(4) a. Base: A word is a syntactic object.
b. Induction: {@,{L,K}} is a syntactic object,
for L and K syntactic objects and @ a label.

(4a) speaks for itself, although it is not totally innocent. The general instance is not too complicated--a word is a lexicon item. However, the Minimalist program allows for the formation of complex words, whose internal structure and structural properties are not determined by the syntax. (Indeed, the object we obtain after Spell-out would also qualify as a word, in the technical sense of having a label and a structure which is inaccessible to the syntax.) (4b) is obtained through Merger, and involves a labeling function which Chomsky argues is necessarily projection. What is relevant to us, though, is how a label is structurally expressed:

(5) Within a syntactic object, a label @ is not a term.
(6) K is a term if and only if (a) or (b):
a. Base: K is a phrase-marker.
b. Induction: K is a member of a member of a term.

(6a) hides no secrets; in turn, (6b) is based on the sort of object that is obtained through merging K and L: a set containing K and L as members, and another set containing {L,K} and the label @; namely, {@,{L,K}}. This whole object (a PM) is a term through (6a). The members of the members of this term (namely L and K) are also terms, through (6b). The label @ is a member of the first term, hence not a term, all directly as desired.

Consider next the linearization of {@,{L,K}} as {@,<L,K>}, equivalent to {@,{{L},{L,K}}}. Through (6b), L and {L,K} are terms. However, {L,K} is not a syntactic object, through either (4a) or (4b), and therefore {@,{{L},{L,K}}} cannot be a syntactic object through (4b), and must be one only through (4a)--that is, as a word. This is well, for we want the linearized object to be like a compound, that is essentially a word: it has a label, it has terms, but it is not an object for the syntax to access.

The reader may appreciate the fact that we have taken our set-theoretic notions very seriously, including the expression of such notions as linearity without any coding tricks (angled brackets, as opposed to particular sets). In essence, we are finding that generally merged structures (those that go beyond the head-complement relation) are essentially and fundamentally non-linear, to the point that linearizing them literally destroys their phrasal base. This conclusion lends some credibility to Chomsky's conjecture that Merge produces a completely basic and merely associative set-theoretic object, with no internal ordering.

Finally, observe how we have now deduced (1b). This statement stipulates that the elements dominated by $ in a command unit precede whatever $ precedes. That $ should precede or be preceded by the other elements in its command unit has been deduced in the previous section. As to why the elements $ dominates should act as $ does within its command unit, this is a direct consequence of the fact that $ has been spelled-out separately from the command unit it is attached to, in a different derivational cascade. Quite literally, the elements dominated by $ cannot interact with those that $ interacts with, in the 'mother' command unit. Thus, their place in the structure is as frozen under $'s dominance as would be for the members of a compound $ or the syllables of a word $.

There is one final, important assumption that I am making which should be pointed out. The 'smart' situation that we have considered can be schematized as follows:

```
(7)          XP
              ^
              |
         MERGE--> X'
              |       /\
  <--SPELL OUT--YP   X ...
            /\
            Y ...
```

But what prevents a 'stupid' projection, as follows?

```
(8)          YP
              ^
              |
          MERGE--> X'
                |      /\
  <--SPELL OUT--Y'   X ...
          /\
          Y ...
```

Note that in (8) it is the spelled-out category Y' that projects a YP. In effect, this results in forcing the linearization of the X projection prior to that of the Y projection, contrary to fact.

The problem is familiar. Both Kayne and Chomsky are forced into ad-hoc solutions to avoid this sort of undesired result involving specifiers (in systems with a single Spell-out). Kayne has to eliminate the distinction between adjuncts and specifiers to avoid the pitfall,[5] and Chomsky has to define command in a peculiar way: only for heads and maximal projections, although intermediate projections must be 'taken into account' as well.[6]

I admit that my solution is also stipulative: (8) can be prevented if only lexicon items project. Multiple Spell-out is designed to turn a phrase into a word compound of sorts. Yet, this "word" cannot be seen as an item that projects any further; it can merge to something else, but never be the item that supports further lexical dependencies. There are several ideas that come to mind in order to make this stipulation more palatable, but I will leave it as such in order not to pre-judge an issue that seems to me rather deep: why words exist with combinatorial properties.

III Some Predictions for Derivations

We have shown how the base step of the LCA follows from economy considerations, and the induction step follows from a Minimalist architecture of the grammar that makes central use of multiple Spell-out, thus yielding a dynamically bifurcated access to the interpretive components. This architecture makes some direct predictions, first of all, given the centrality of the emergent command units. In a nutshell, command is important in the grammar because it is only within command units that syntactic terms 'communicate' with each other, in a given derivational cascade.

To get a taste of this sort of prediction, consider the notion of distance that Chomsky (1995:chapter 4) discusses, which is sensitive to command. The reason to involve command in the characterization of distance is empirical, and has to do with 'superiority' effects of the sort in the paradigm below:

(9) a. Who t saw what?
b. * What did who see t?
c. Which professor t saw which student?
d. Which student did which professor see t?

Chomsky's account capitalizes on the fact that the competing Wh-phrases (who, what, which) stand in a command relation in (9a/b), as seen in (10a), but not in (9c/d), as seen in (10b):[7]

(10) a. [C [who...[saw what]]]
b. [C [[which professor]...[saw [which student]]]]

Thus he defines distance in terms of the proviso in (11):

(11) Only if @ commands & can @ be closer to $ than & is.

This is the case in (10a): the target C is closer to who than to what. Crucially, though, in (10b) the target C is as close to the which in which professor as it is to the which in which student, and these positions being equidistant from C, both movements in (9c) and (9d) are allowed, as desired. Needless to say, this solution works. But why should this be? Why is command relevant?

In our terms, the explanation is direct. The two Wh-phrases in (10a) belong to the same derivational cascade, since they are monotonically assembled through merge into the same PM. This is not true about the two Wh-phrases in (10b); in particular, which professor is assembled in a separate command unit from that of which student, and hence they do not really compete within the same derivational space. The fact that they are equally close to the target C is thus totally expected, and need not be stated in a definition of distance, since it already is architecturally true.

This general reasoning extends to classical restrictions on extraction domains, which must be complements.[8] This is a nasty paradigm for the Minimalist program, as (12) demonstrates:

(12) a. [... X [...t...]] e.g. Who did you see [a critic of t]

<-------'

b. [[...t...] X ...] e.g. *Who did [a critic of t] see you

<--*--'

The problem is that whatever licenses (12a) in terms of the Minimal Link or Last Resort conditions should also license (12b); so what is wrong with the latter? A minimalist cannot simply translate the observation in (12) into a new principle; such a principle must again fall within the general desiderata of the program--and thus reduce to economy or bare output conditions. I know of no minimalist way of explaining the contrasts in (12).[9]

But now consider the problem from the present perspective. A complement is very different from any other dependent of a head in that the elements a complement dominates are within the same command unit of the 'governing' head, whereas this is not true for the elements a non-complement dominates. As a result, extraction from a complement can occur within the same derivational cascade, whereas this is not possible for extractions from a non-complement, given our assumptions. Basically, we find the following paradox. If we spell-out a non-complement independently from its head, any extraction from the non-complement would involve material from something which is not even a syntactic object--thus should be as bad as extracting part of a compound. On the other hand, if we do not spell-out the non-complement in order to allow extraction from it, then we will not be able to linearize its elements, always assuming that our only procedure for linearization is the command-precedence correspondence that economy considerations sanction.

Of course the reader may now wonder how we can generate such simple structures as (13), with movement of a complex Wh-phrase:

(13) [[which professor] [did you say [t left]]

If to sanction the linearization of a complex non-complement's elements we must spell them out prior to their merger to the rest of the PM, how can movement of non-complements exist? Shouldn't they be pronounced where they are spelled-out, as in (14)?

(14) * [did you say [[which professor] left]]

Indeed this "should" be the case, predicting only in-situ Wh-phrases in non-complement position. This would be the end of the theory if some subtle issues didn't arise when dealing with chains.

The main question is whether movement, which involves copying some material and then merging it, must have these two steps immediately feeding one another, or rather they are independent, even if ultimately related. If the latter is an option, movement of a complex phrase may proceed in several steps, as follows:

(15) (i) Copy one of two independently merged command units:

```
L      L     K
/\<----/\   /\
...COPY...   ...
```

(ii) Spell-out as trace the lower copy:

```
L    L    K
/\  [0]  /\
...       ...
```

(iii) Merge the trace:

```
       N
       /\
 L    L  K
/\   [0] /\
...      ...
```

(iv) Merge the higher copy:

```
          M
          /\
         L ...
         /\  \
 ...  N
            /\
           L  K
          [0] /\
               ...
```

The key to the whole derivation is the sort of 'parallel' movement implied in (15i-ii)--the rest of the steps are straightforward. So let us see whether the initial steps can be justified.

Technically, what is implied in (15i-ii) is basically the same as what is involved in the formation of a PM as in (2a):

(16) Numeration {the, a, man, saw, woman, ...}

```
   the          saw
   /\           /\
 the man      saw  a
                   /\
                 a  woman
```

Prior to merging [the man] and [saw [a woman]] we must assemble them in separate derivational spaces. (15i) capitalizes on this possibility; instead of copying lexical items from the numeration, as in (16), in (15i) we copy the items from the assembled PM. Only an unmotivated stipulation would prevent this process.

In turn, (15ii) employs the option of deleting phonetic material, thus making it uninterpretable for PF interpretation. It is reasonable to ask why this step is involved, but the question is no different from that posed by Nunes (forthcoming), concerning why the copy of K is not pronounced when K is moved:

(17) (i) [K...[...K...]...]   (ii) [K...[...[0]...]...]
^_ _ _ _/

We can adapt (any version of) Nunes's answer directly to why (15ii) is taken. As a result, a chain (L, L) may be formed in (15).

Technical considerations aside, though, Nunes (1995) offers a very interesting argument for the parallel movement involved in (15). He analyzes parasitic gaps this way:

(18) a. What did you file after reading
b. (i) Merge the adjunct:
[after reading [what]]
(ii) Copy the Wh-phrase:
[what]   [after reading [what]]]
(iii) Merge the copy:
[file [what]] [after reading [what]]]
(iv) Merge everything:
[[file [what]] [after reading [what]]]
(v) Copy first Wh-phrase higher up in the PM:
[what [did you [[file [what]] [after reading [what]]]]]
(vi) Create gaps:[10]
[what [did you [[file [0]] [after reading [0]]]]]

The key step is (ii), a parallel copy in a separate derivational space. By creating parasitic gaps this way, Nunes is able to predict some of their core properties, including why parasitic and real gaps must bear the same Case (they are mere copies),[11] why the Wh-phrase must command both gaps (the condition for gap creation through deletion), or why neither gap can command the other (or a chain would automatically be formed between the two, with inappropriate thematic properties).

As far as I can see, parallel movements are a necessary property of the multiple Spell-out system, or non-complements would never be moved. On the other hand, we must now make sure that we have not allowed extractions from inside non-complements, once we have simplified the system to allow the sort of derivation sketched in (15). The most problematic instance I can think of is (19):

(19) a. * [who did [[a critic of t] see you]]

b. (i) Copy part of one of two separate command units:

```
      L    K
      /\   /\
  J  ... J...   ...
  ^_ _ _ _/
```

(ii) Spell-out as trace the lower copy:

```
      L    K
      /\   /\
  J  ...J...  ...
 [0]
```

(iii) Merge the command unit containing the trace:

```
       N
       /\
  J  L    K
     /\   /\
   ...[J]... ...
```

(iv) Merge the higher copy:

```
        M
        /\
        J ...
            \
             N
             /\
  L    K
  /\   /\
        ...[J]... ...
```

(19biv) is in effect the structure corresponding to (19a)--so we must prevent the derivation in (19b). Again, as minimalists we must do so without adding further machinery to the system.

One straightforward way of preventing the unwanted derivation is in terms of the structure that results from merging the command unit containing the trace, (19biii). The logic of our system forces partial spell-out of the unit containing the trace, or the larger structure will not be linearizable. Upon linearizing L in (19), its internal structure becomes inaccessible to the syntactic system, since the spelled-out phrase is no longer a syntactic object. As a consequence, we cannot determine the existence of a chain (J,J), since the lower copy is lost in the word dimension of the spelled-out L. (This is all assuming, of course, that the determination of a sequence of copies as a chain is a syntactic process.) In contrast, (15) faces no such problem, since it is L itself, in that instance, that must form a chain with its upper copy. The internal structure of L is lost, but not L, as desired. Hence forming a chain between the upper and lower L's is as simple as it would be to form a chain between a moved word and its (silent) copy.

IV General Predictions for the Interpretive Components

The dynamically split model that multiple Spell-out involves produces derivational cascades, each of which reaches the interpretive components in their own derivational burst. If this model is correct, we should see some evidence of these bursts.

For reasons of space, I will only sketch the general shape of what we expect to find, commenting on some obvious difficulties. Let us start with PF matters. Perhaps the first sort of prediction that comes to mind relates to recent work by Cinque which goes back to Chomsky's early observations on focus 'projections'. Generally speaking, the focus that manifests itself on a 'right branch' may project higher up in the PM, whereas this is not the case for the focus that manifests itself on a 'left branch'. For instance:

(20) a. Michael Angelo painted THOSE FRESCOES.
b. MICHAEL ANGELO painted those frescoes.

(20a) can be an answer to several questions: 'what did Michael Angelo paint?', 'what did Michael Angelo do?', and even 'what happened?'. In contrast, (20b) can only answer the question 'who painted those frescoes?'. What does this asymmetry follow from?

Our architecture is very consistent with it, regardless of what is the ultimate nature of focal spreading. The main contribution to the matter that we can make is evident: for us focus can only spread within a command unit--that is, up a 'right branch'. Spreading up a 'left branch' would be moving across two different command units, and hence an instance of a cross-dimensional communication between different elements.

There are a variety of other phonetic domains that may well conform to this picture: those having to do with phrasal stress, intonational phrasing, pausing, and others. The truth is, however, that I am not prepared to explore these matters seriously beyond pointing out what is obvious, as I did for focus spreading. In general, 'left branches' should be natural bifurcation points for phonetic processes, if the present architecture is on target.

My interests lay more within the LF component, where there are both immediate predictions and also notable problems, which force us to explore new possibilities. Once again, the general prediction is obvious: command units should be natural domains for LF phenomena. And of course this is true for a variety of processes, from binding of different sorts to obviation, from scopal interactions to negative polarity licensing, and so on; it is much harder to find instances of LF processes that do not involve command than otherwise. Needless to say, though, there are such instances--some of which I return to. But more importantly, we must observe that command units are just a sub-case of the situations where command emerges in standard structures.

A problematic instance emerges in a now familiar structure:

```
(20)        M
            /\
            J ...
               \
                N
               /\
             L    K
            /\    /\
          ...H... ...
```

Although J and H are not part of the same command unit (the latter is part of a command unit dominated by L) it is certainly the case that J commands H. Empirically, we want the relation of (20) to take place in situations of antecedence, where J tries to be H's antecedent (as in sometimes, everyone thinks that [his parents] hate him). So the question is simple: if J and H are in different syntactic dimensions--after spelling-out L--how can J ever relate to H? The situation is quasi paradoxical, since we used this very reasoning to prevent chain formation in (19).

The logic of the system forces an interesting answer: unlike chain determination, which is derivational (together with copying, deletion, and the rest), there are aspects of the notion of antecedence which are irreducibly non-derivational. This might mean that antecedence is a semantic or pragmatic notion; either way, we are pushing it out of the architecture seen thus far--at least in part. The hedge is because we still want antecedence to be sensitive to command, even if it does not take place within the command units that determine derivational cascades. As it turns out, the dynamically split system does have a bearing on this.

Essentially, we want to be able to say that J in (20) can be the antecedent of H, but not conversely. Now, recall that although L in (20) is not a syntactic object after Spell-out, it does have internal structure, and its information is not lost. To see this in detail, suppose that L had the internal structure, prior to Spell-out, of his parents, that is: {his,{his,parents}}. After Spell-out, the structure becomes {his,<his,parents>}, equivalent to {his,{{his},{his,parents}}}. Through (6b), we can determine his and {his, parents} as terms, an important fact because, although the linearized object is not a syntactic object, it has terms within it, which the system can identify even if not operate with, given that they do not constitute a licit structure.[12] The point is this: if the relation of antecedence is based on the identification of a term like his, even the linearized structure does the job, frozen as it is for syntactic operations.

But consider the converse situation, where in spite of H being a term in (21), it cannot be the antecedent of L or K:[13]

```
(21)        M
           /\
          J ...
         /\  \
       ...H... N
        /\
       L   K
```

This suggests a characterization of '@'s antecedent' as in (22):

(22) Where @ is a term in a derivational cascade D, a term & is @'s antecedent only if & has accessed the LF component in D.

This determination of '@'s antecedent' is completely derivational, unlike the notion 'term' in (6), which is neutral as to whether it is characterized derivationally or representationally. It is worth emphasizing that (22) is not attempted as part of a definition of antecedence, but rather of antecedent of @.[14] In any case, the formal aspects of the notion in (22) are compatible with and indeed natural within the system presented thus far, but its substantive character--the fact that only a term that accesses LF in D can be an antecedent of the terms in D--does not directly follow from the architecture. Differently put, the substantive character of (22) is, as far as I can see, irreducible.

I do not want to finish this section without placing it in a larger perspective, for empirical reasons internal to a Minimalist architecture. Much of what I have just said must be qualified, in light of the puzzle posed by the paradigms in (23) and (24):[15]

(23) a. [And the fact is [that there is a monk in the cloister]]
b. [And there is the fact [that a monk is __ in the cloister]]

(24) a. [A ball-room was [where there was a monk arrested]]
b. [There was a ball-room [where a monk was __ arrested]]

These two paradigms come from the same numerations. Curiously, watching the embedded clauses we immediately see that the (b) examples involve a local movement of a monk, where an option of inserting there exists which the (a) examples take. Given the assumptions in Chomsky 1995:chapter 4, the (a) examples should outrank the (b) examples, but in fact both are grammatical.

A solution to this puzzle is evident: we must split the derivations of the examples in (22)/(23) in something more drastic than two separate cascades for the embedded and matrix clauses. In fact, we need two entirely separate derivational flows, each taking stock from a separate numeration. If this is granted, the embedded clauses in each pair do not compete, since the numerations are different (crucially, there falls in each of the two separate numerations involved in each example). Then of course we need to assemble a syntactic structure through some paratactic process.[16]

None of these ideas is particularly new. I am suggesting a return to something very much like the 'kernel sentences' of the fifties, and the seasoned reader will no doubt see various sources for the paratactic analysis that needs to be invoked. The point is, all of this will change the rules even more drastically than I have suggested, involving not only different derivational cascades, but as I have said also entirely separate derivational flows. Indeed, parataxis entails a form of discursive reconstruction of sorts, which forces us to look beyond standard sentence grammar. This is all to say that the situation sketched in (20)/(21) may well fall outside of the standard confines of what we can think of as the kernel grammar, going into a higher-order grammar where such matters as discourse files, and their determination and access, could be very relevant. From this perspective, it could well be that (22) must be established not so much for derivational cascades, but rather for derivational flows; it might even be part of the paratactic characterization, as a sort of 'coordination' condition to assemble the components of separate kernel sentences, super-merged at the level not just of categories, but of whole derivations.[17] But these are just matters to explore.

Nonetheless, one immediate consequence of the suggested, transderivational architecture, is to leave standard LF movement untouched, as much as possible. That is, consider sentences like [the fact that there was a riot in the town/the town rioted] lead to some radical changes. The issue here is what happens inside the 'left branch', in particular the bit there was a riot in the town/the town rioted. By hypothesis, these derivations involve feature movement at LF, for familiar reasons having to do with Case. I don't have any intentions of questioning those standard analyses, but if I go into early Spell-out with these sub-structures, their phrasal character will be destroyed, turning them into compounds of sorts. Then it is not clear what it would mean to say that we can move items within these flattened structures, anymore in the covert components than in the overt ones.

However, if the separate clauses there was a riot in the town/the town rioted are assembled in altogether parallel derivations (and not just parallel cascades within the same derivation), then we can do whatever it is that derivations do in these structures, all the way down to LF movement. This still predicts (in fact a more radical) islandhood of these clausal 'left branches', and leaves the analysis of non-complement nominals untouched. In turn, if the only way kernel numerations are assembled is around truly clausal structures--not nominal or other 'defective' projections--then we should not find expletive replacement or standard structural Case marking inside nominal structures, as is obviously the case.[18]

V Beyond Derivations

There are a variety of apparently LF phenomena which do not fit the Minimalist architecture, but are reasonably seen from the point of view of parataxis and the mechanism to associate the outputs of totally separate derivational flows (as opposed to derivational cascades coming from the same flow). I am mainly thinking of Weak Crossover, which is not predictable from the present architecture.

In brief, I need to think of the phenomenon as taking place in the mapping from LF to whatever comes beyond, which now I am suggesting may be (at least in part) a Paratactic Component. It is natural to think of such information theoretic conditions as Wasow's Novelty precisely in this light, and it may well be that Crossover effects reduce to unacceptable instances of novel access to familiar expressions.[19] I cannot go into any such analysis here--but I am forced into it, or something very much like it.

However, we must ask an even more basic question, given all this talk on what does or does not go into LF or PF: What evidence do we have that we need PF and LF levels? Surely, conceptual necessity alone forces on us PF and LF components of the system, but what does it really mean or buy us to say that, aside from the conceptually necessary, we also have a level to represent it?

Perhaps we should start with a careful determination of what we mean by a level of representation. In a non-technical sense, a 'level' just means a coherent collection of phenomena obeying some conditions; since we have a technical definition as well, I propose that we keep the term component to refer to this informal notion. The technical notion comes from the theory of concatenation algebras, and right away we should have a problem with this. As I have explicitly insisted, the Merge operation yields a much looser dependency among terms than concatenation--which codes linear ordering. This is no trivial point; this article is meaningless if we assume a concatenation algebra for syntactic representation.

Nonetheless, consider the standard definition:[20]

(25) A level of representation L is a system which presupposes:
A. A set of primes from a substantive alphabet.
B. The operation of concatenation, forming strings of primes of arbitrary length.
C. Some specific relations among the terms thus defined.
D. A designated class of (sets of) strings of primes called 'L-marker'.
E. A mapping M of L-markers onto L'-markers, generating an implication L>L'.

Again, we lack the crucial (B), but suppose this is fixed by extending our notions appropriately from concatenations to mere associations (not coding order). Can we still say that, in particular, LF is a level in the sense in (25)?

LF is not mapped anymore from any other level such as S-structure or D-structure, in satisfaction of (E). This is one of the main differences between Minimalism and previous models; where the Government and Binding system would generate I seem to have been arrested as in (26), Minimalism proceeds as in (27):

(26) a. D-structure: [seem [to have been arrested I]]
b. S-structure: [I seem [t to have been arrested]]

(27) a. Partial output of Merge: [to have been arrested I]
b. Partial output of Move: [I [to have been arrested t]]
c. Partial output of Merge: [seem [I [to have been arrested t]]]
d. Partial output of Move: [I [seem [t [to have been arrested t]]]]

(26) involves levels that we have now rejected, mainly for conceptual reasons.[21] But the outputs of each operation in (27) are interleaved, in such a way that we cannot coherently define an L-marker in the course of the derivation. Thus, (27a) involves only categories arranged into PMs; (27b) already involves some different objects: chains. (27c) has both chains and some more categories in the upper chunk; (27d) has some new chains. It is only after the LF component is done with its work that we may have a single, coherent collection of chains, the LF level. Similar issues arise in the PF component, with whatever units are relevant after we are done with PF

operations, to generate the PF level.

It is perhaps worth emphasizing that chains and categories are indeed very different creatures. A category is a configurational notion, and is central in determining such issues as thematic properties. In contrast, a chain is a supra-configurational notion, a relation among configurations which is mediated through a moved element. The moved element can be mistaken with the chain links, because of notational matters that should probably be clarified:

(28) [K [...[L (K)]...]]
^_M_ _ _ _/

(I parenthesize the copy of the moved K, which is to be interpreted as K's trace.) The chain formed in (28) is not {K, K}, but is instead {M, L}. Note that {K, K} could not be an object in the system unless we introduce some notational trick, for the very simple reason that the two occurrences of K are identical. The obvious way to distinguish the upper from the lower K is in terms of their phrasal contexts: {{K,M},{K,L}}; however, this is informationally identical to {M,L}. In any of these formats, though, the chain is obviously not a syntactic object in the sense of (5). This is well, since we want categories to be syntactic objects that the syntactic derivation operates with, whereas chains are the output of this derivation, which the semantic representation is eventually built from.

These matters illustrate themselves in the formation of complex chains, as in (27). The output of Move in (27b) creates the chain {{I,to},{I,V}}, equivalent to {to,V}. Again, this is not a syntactic object, which means we cannot operate with it and further movements must instead work with something like the categorial I, a syntactic object by way of (5a). This is what we see in (27d), where a new chain {{I,Infl},{{I,to},{I,V}}} is created, equivalent to {I,{to,V}}--not a syntactic object either if we restrict labeling to projection. The LF level is composed of articulated collections of chain objects like {to,V}, {I,{to,V}}, and so on, which serve as the base for some semantic representation of (A-moved) arguments and (head-moved) predicates, including the complex predications involved in (A'-moved) quantificational elements.

Now: do we really need the uncontroversial collection of chains to be a level, specifically?[22] Indeed, an even more basic question can be posed: can we simply state the fact that the LF collection of chains constitutes a level? Of course, we meet with this collection the trivial requirement A in (25); but we would too even if we only had an LF component--so this is not a deciding factor. And again, the implication in E and the structuring function in B are not met, but they are arguably not crucial either. As for D, we will not find sets of strings, but we do indeed find sets of whatever alternative structuring we propose to substitute for concatenation. Finally, there is C, a set of relations among the chain associations. At first, it seems as if this defining condition is also trivially met, but this is before we think of what exactly unifies the set of chains where these relations (the basis for semantic structuring) are definable.

Take a numeration such as {the, a, man, saw, woman, ...}, as in (16), which allows us to construct the two separate objects [the man] and [saw [a woman]]. Why do we ever assemble these two objects? Three answers are in principle possible. One answer is derivational. For instance, in the case of English one could say that an unwanted derivation results in a strong feature in Tense not being checked (by [the man]), leading to a cancellation. The problem with this sort of approach is that it might not be general enough; first, what about a language with no such feature?; second, even if all languages had strong Tense features, the general problem goes well beyond subjects and predicates: the issue is what forces just any composition of two elements--we need to posit a strong feature every time we want a combination of symbols.[23]

Another answer to the general problem is representational: not assembling subject and predicate leads to a violation of Full Interpretation. This general answer is not really straightforward if LF, in particular, is a collection of chains. Why exactly should a chain system care about a combinatorial condition <u>on categories</u>? We should bear in mind that in the old days the combinatorial conditions were imposed on D-structure, which served as a unifying level, which other levels were built from. Now that D-structure is gone, if we want to impose a categorial (D-structure) requirement on a chain (LF) level, we want to have the cake and eat it too. Of course, one could always try to argue that the combinatorial requirement is not categorial or even configurational, but a chain requirement. However, that will face empirical difficulties with such examples as (29), where a chain (not a category) gets a role:

(29) a. [I [t believe [t to be [a monk in the cloister]]]]
 ^__/^_____/

b. [I [a monk [t believe [t to be [t in the cloister]]]]]
 ^_________________/

Observe that <u>I</u> moves first to the position where it can receive a subject role from <u>believe</u>, and then to subject position where it checks nominative Case. In the LF component, <u>a monk</u> moves to the position where it checks accusative Case, over the traces of I (hence in satisfaction of the

Minimal Link Condition). The only way to rule out the ungrammatical (29) is if by preventing movement to get a role (cf. the perfect I believe there to be a monk in the cloister). Of course, this is just what D-structure ensured: in that configurational level, roles were assigned by definition, and nowhere else. D-structure may be gone, but not its basic effects.

A third answer to the combinatorial problem is purely interpretive: if we do not combine, say, [the man] and [saw [a woman]], the result is uninterpretable. This seems like the most straightforward answer, although it does pose a couple of non-trivial difficulties. First, we still do not have a scientifically falsifiable theory of interpretation. Second, a convergent, uninterpretable derivation may have the nasty effect of ruling out an interpretable one which is a bit more costly; this is because (un)interpretability does not enter the computation of derivational optimality. Thus, take the derivation leading to [[the man] [saw [a woman]]] and that leading to [the man]...[saw [a woman]]. Which is simpler? The question is fair, since both derivations come from the same numeration, and may share partial numerations as they are being reduced to zero--so the comparison is legitimate at the numeration end. It is also a legitimate comparison at the LF end, since by hypothesis both derivations converge. Then again the good news is that one is not obviously more costly than the other: although one derivation involves one less step of Merger, the unmerged alternative involves one more step of Spell-out. In any case, these issues should be raised when pondering the alternative of merging a unified LF object for interpretive reasons.

The reader may have noted that the simplest answer to the unification/composition puzzle does not imply a level of (in this case) LF; since the unification of the syntactic object is done for interpretive reasons, there is no obvious need for a level of LF where the entire set of chains that feeds interpretation is previously unified. Interestingly, the multiple Spell-out system also lends itself naturally to this conclusion. Spelling-out multiply entails accessing the LF/PF components in a dynamic way, and after this dynamic access, it is not obvious why matters should ever become more static, 'waiting', as it were, for the assembly of a single level of representation. The obvious alternative is to feed the interpretive mechanisms as the derivation unfolds.

Needless to say, whether this conclusion is correct is an empirical matter, and one that I will not even remotely attempt to assess at the end of this article. I simply want to raise the question, and point out some obvious facts. First, the emerging system would be much more radically derivational than its present alternative. Both the fact that the system would be derivational, and radically so, are interesting ideas on their own right. Second, the radically derivational system would have no place for the Principle of Full Interpretation as a syntactic condition (although it would be a possible interpretive requirement). In the present system derivations can crash (when violating Full Interpretation) or be cancelled (when violating derivational requirements, such as cyclicity); the radical alternative would simplify this picture, since derivations could not crash: there would be no level for them to converge to. Instead, derivations would be well-formed or cancelled--the task remains of rethinking the empirical results of crashing as derivational cancellations. Third, not having a PF or LF level takes us closer to the interpretive components of the system, which in the Minimalist program are, by definition, performative; that is, the net result of multiple Spell-out is, in part, getting us closer to linguistic performance.

Notes

1. The contents of this paper have been presented in several lectures, at the universities of Connecticut, Delaware, Maryland, Oxford, Stuttgart, Potsdam, and Rio de Janeiro, the City University of New York, the Max Planck (Berlin), San Raffaelle (Milan) and Ortega y Gasset (Madrid) institutes, and the School of African Studies in London. The generous hosts of these institutions, as well as the audiences, are warmly welcomed for comments, questions, and criticisms. I am especially indebted to Stephen Crain, Jim Higginbotham, Howard Lasnik, and Jairo Nunes, for useful commentary, and Maggie Browning for her interest in this research.

2. For instance, in various works by Jackendoff, Lasnik, and Bresnan, now partly outdated.

3. Although we have also used the concept of information economy in the previous section. Clearly, a single application of Spell-out is involves more symbols than each of the applications of multiple Spell-out. Furthermore, it can be shown that the sum of the partial entropies (information cost) involved in multiple Spell-out is less than the total entropy of single Spell-out, if we consider the possible relations among the symbols involved in each instance. Thus, since we do want to spell-out singly whenever possible, we must relativize entropic cost to standard derivational cost, in that order. I simply state this fact, without pursuing it.

4. I am assuming the standard mathematical definition of a sequence <a, b> as a set {{a}, {a, b}}.

5. If specifiers are adjuncts, one can then blame their being linearized prior to corresponding heads in the poorly understood concept of adjunction.

6. If command is not defined for an intermediate projection, this category will never command (hence precede) its specifier. The converse is true by fiat, given that a specifier is by definition a maximal projection. At the same time, intermediate projections must be relevant in computing command, or else a head and its specifier would command, hence precede, each other.

7. See Kitahara 1994 for these ideas.

8. The presentation that follows owes much to useful discussions with Jairo Nunes and, as should be apparent, to Nunes 1995.

9. Ormazabal, Uriagereka, and Uribe-etxebarria 1994 and Takahashi 1994 do have similar proposals about the matter, which prevent extractions from inside subjects in terms of the Uniformity Condition. However, neither of these proposals extend to extractions from inside adjuncts, unlike the present proposal (assuming adjuncts are non-complements). Needless to say, this view of the world is contra the spirit of Larson 1988.

10. I have ordered this step the last for expository purposes. If it involves deletion, though, it may be cyclically ordered. Evidently, this implies a certain look-ahead in the system. Nonetheless, the matter is extremely local if the relevant displacement is successive cyclic. As a matter of fact, the need for local look-aheads may be what predicts an otherwise surprising successive cyclicity in this process of long displacement. The matter extends directly to the analysis of subject and adjunct Wh-movement that is being attempted in this section.

11. In languages where this condition does not hold, such as English, it can be assumed with Lasnik 1996 that the numeration does not code morphological features; instead bare forms are used.

12. Interestingly, parents does not come out a term. In fact, the reader can easily see that no 'last' element in a right branch ever comes out a term after a PM is linearized. (Of course, prior to that, these elements are normal terms.) Technically, this entails that said elements cannot have an antecedent, which if pushed to a logical extreme might well mean that they cannot have reference. This would lend itself nicely to the idea, expressed in Uriagereka 1996, that the 'last' element in a right branch is always the predicate of a small clause. I cannot pursue the matter here.

13. At least, it is not obvious that an antecedent buried inside a 'left-branch' can hook-up to a variable in a different command path. There are apparent exceptions, such as (i):

(i) ? Everyone's mother likes him.

To the extent that these examples are good, though, they may well involve a process akin to, but formally unlike, variable binding. If pronouns like him in (i) can be analyzed as incomplete definite descriptions, then we are really dealing with a version of (ii):

(ii) Everyone's mother likes the one that is relevant.

In effect, by co-operatively confining the range of the context variable of him, we may end up with a semantics which is truth-conditionally equivalent to that implicit in (i). Then the question is what conditions obtain for context confinement, something which need not be sensitive to the strict command restrictions that are presently being explored for the syntax.

14. I mention this to address a reasonable objection that Jim Higginbotham raises: semantically, it makes sense to say that 'an anaphor seeks an antecedent'; but what does it mean to say that 'an antecedent seeks an anaphor'? The point is well-taken, but we are not dealing with semantics here. The question, really, is whether it makes syntactic sense to define the notion 'antecedent of'. Note that in parsing this is certainly a reasonable question to raise: it is generally antecedents that the parsers finds, only then proceeding to worry about corresponding anaphors. To be fair, though, this is completely true only in terms of fillers and gaps, and in any case it may be that parsing considerations should not be taken into account when dealing with competence decisions. Be that as it may, it is not immediately incoherent to stipulate that an antecedent (in Higginbotham's terms, the 'head' of a linking arrow) is something which must be syntactically accessible within a given derivational cascade. One way to think of the process is thus: the system actually 'knows' it has stored an anaphor of some sort, as the derivation proceeds (regardless of whether the command unit containing this anaphor has been 'cashed out' at partial Spell-out); then at some point in the derivation the system finds out a valid (that is, syntactically accessible) antecent for the anaphor, and the relevant link is established.

15. The example in (23) was pointed out to me by Juan Romero.

16. Of course, we must also prevent this sort of analysis for Chomsky's (i), the main reason we assume 'numerations':

(i) a. *There was believed [a man to be t here].
b. There was believed [t to be a man here].

(ib) outranks (ia). But if we allow (ia) to be the result of two separate derivations paratactically related, as in (ii), there is no way (ib) can outrank it, just as (23b) does not outrank (23a):

(ii) there was belived it
^ [a man to be here]

One straightforward way to prevent (ii) is that the parallel derivations are both in violation of Case theory. It is only if we allow LF movement after the derivations are paratactically assembled that (ia) would satisfy Case requirements. The trick is, then, to make the natural assumption that paratactic dependencies are "post cyclic", without any syntax across them.

17. I have in mind something along the lines of Chierchia's notion of 'dynamic binding'.

18. More accurately, we should not find these processes in non-complement nominals and related structures. Nothing I have said predicts why (i) is ungrammatical:

(i) *I know [there destruction of Rome].

19. One evident instance of this is (i):

(i) His friend killed a man.

The familiar expression his friend cannot antecede the novel a man. In order to extend this analysis to the examples in (ii) we must postulate that the relevant operators involve an existence predicate of the sort postulated by Klima in the early seventies, which gives them an indefinite character;

(ii) a. His friend killed everyone.
b. Who did his friend kill.

That is, the logical form of everyone must be similar to its morphological form: Every x, one(x); and something similar must be said about who, a matter which is well-known for various languages (see for instance Kim 1991, or Watanabe 1992).

20. For details, see Chomsky 1955.

21. There is also an empirical argument against D-structure, posed by Kevin Kearney. It has to do with the grammaticality of (i):

(i) A man who is easy to please is easy to convince.

For reasons discussed in Chomksy 1981, the subject of a 'tough-construction' cannot be base generated; the problem is that the subject in this instance is a whole clause, furthermore one involving a 'tough-construction'. Hence we get into a regress that renders the notion D-structure vacuous.

22. I really mean the word 'uncontroversial'. Some other systems may not make use of chains, per se, or may not define them as above. Nonetheless, whatever notational alternative one may attempt will simply restate one of the more basic facts of human language: categories are crucially displaced, for some obscure reason.

23. See Hoffman 1996 for an exploration of one such system.

References

Chomsky, N. (1955) The Logical Structure of Linguistic Theory, published in 1975, Chicago: Chicago University Press.
Chomsky, N. (1981) Lectures on Government and Binding. Dordrecht: Foris.
Chomsky, N. (1995) The Minimalist Program. Cambridge: MIT Press.
Epstein, S. (forthcoming) 'Unprincipled Syntax and The Derivation of Syntactic Relations.' To appear in M. Browning, ed., Working Minimalism. Cambridge: MIT Press.
Hoffman,J. (1996) Syntactic and Paratactic Word-order Effects. Doctoral dissertation, University of Maryland.
Kayne, R. (1994) The Antisymmetry of Syntax. Cambridge: MIT Press.
Kim, S. W. (1991) Scope and Multiple Quantification, Doctoral dissertation, Brandeis University.
Kitahara, H. (1994) 'Deducing Superiority Effect from the Shortest Chain Requirement,' in Harvard Working Papers in Linguistics,3, Harvard University.
Larson, R. (1988) 'On the Double Object Construction.' Linguistic Inquiry 19:335-391.
Lasnik, H. (1996) 'Verbal Morphology: Syntactic Structures Meets the Minimalist Program.' In H. Campos and P. Kempchinsky, eds., Evolution and Revolution in Linguistic Theory.Essays in Honor of Carlos Otero. Washington D. C.: Georgetown University Press.
Nunes, J. (1995) The Copy Theory of Movement and Linearization of Chains in the Minimalist Program. Doctoral dissertation, University of Maryland.
Nunes, J. (forthcoming) 'On Why Traces Cannot be Phonetically Realized.' To appear in M. Browning, ed., Working Minimalism. Cambridge: MIT Press.
Nunes, J. and E. Thompson (1997) Formal Appendix to A MinimalistDialogue on Human Language, by Juan Uriagereka. Cambridge: MIT Press.
Ormazabal, J., J. Uriagereka, and M. Uribe-etxebarria (1994) 'Word Order and Wh-movement: Towards a Parametric Account,' talk presented at GLOW 17, Vienna.
Takahashi, D. (1994) Minimality of Movement, Doctoral dissertation, University of Connecticut.
Uriagereka, J. (1988) On Government. Doctoral dissertation, University of Connecticut.
Uriagereka, J. (1996) 'Warps: Some Remarks on Categorization.' University of Maryland Working Papers in Linguistics, 4.
Watanabe, A. (1992) 'Subjacency and S-structure Movement of Wh-in situ,' Journal of Asian Linguistics 1: 255-291.