4-7-2000

From Meaning to Form and Back Again

Michael McOmber

Follow this and additional works at: https://scholarsarchive.byu.edu/dlls

BYU ScholarsArchive Citation
Available at: https://scholarsarchive.byu.edu/dlls/vol26/iss1/19

This Article is brought to you for free and open access by the Journals at BYU ScholarsArchive. It has been accepted for inclusion in Deseret Language and Linguistic Society Symposium by an authorized editor of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
From Meaning to Form and Back Again

Michael McOmber

INTRODUCTION

This paper discusses the advantages of a grammar that places Logical Form as the initial component (Early Logical Form) rather than in the later dovetail position of the Government-Binding model of Chomsky, or after Spell-Out in his Minimalist program of the 90s (Late Logical Form). Chomsky’s interpretivism runs from form to meaning only, whereas the production model I propose here runs from meaning to form for the encoder/speaker and then back again from form to meaning for the decoder/listener. By comparison, Government Binding scrambles half of production.

I set the stage in the first section by rehearsing the relevant principles of Government Binding (GB) and Minimality that I will refer to, in addition to other general arguments which I claim make for good grammar—an overriding principle is the time-honored philosophy of economy in Occam’s Razor, which eschews redundancy and wastage. In the second section I focus in on specific data from Kenstowicz’s analysis of Bani-Hassan Arabic, which includes a crucially overt scope marker. I show that in a Late Logical Form model, the grammar runs into a paradox when it comes time to insert the scope marker. At the crux of the problem is the gnawing fact that, even though interpretation is not supposed to occur until (Late) Logical Form, our grammarian becomes the proverbial “man who knew too much” when trying to achieve some explanatory adequacy on what is happening. That is, he must rely on semantics to explain the insertion, and yet the interpretation to provide it does not occur until after he has completed the step. He essentially knows too much, too soon of the eventual meaning, none of which, in fact, has been computed yet, since it must wait for Late Logical Form to compose the interpretation.

Samples of other fatal consequences of Late Logical Form are presented in the third section, including reconstruction effects, dis-anaphora, and the ironies of defective convergence and the procrastination principle in a grammar that claims its own appeal to “Economy of Derivation and Representation” (Chomsky 1995, chap. 2).

Finally I recapitulate conclusions in the fourth section: Early Logical Form provides more efficiency, therefore higher descriptive and explanatory adequacy. My provocative plea is to abandon interpretivism and pursue a deterministic production model.

THE PREMISES

As this is essentially a rebuttal of Chomskyanism, I begin with the more fully articulated Government-Binding model of his Transformational grammar as a starting point. Although his minimality program promises to be different, all indications are that he has maintained a late interpretive logic there as well. Therefore, what I have to say on Government Binding still holds true of more recent development. The dovetail point at S-Structure acts too much like the Spell-Out point to his new interfaces.
It makes for stronger rebuttal to start with Chomsky's own definitions and constructs (especially as perceived by his adherents) and then show how these lead to contradictions and internal inconsistencies, with his own theory to lead the way, "hoist with his own petard" (Hamlet III:4). Here then follow several guiding principles and premises from Government Binding which we crucially refer to during our discussion.

The Purpose of Logical Form

The function of Logical Form (LF) is to provide a component that is "interpretive in nature [and which] assigns . . . a semantic representation" (Ouhalla 1994, 46) to the various structures which the grammar has generated. It is the "level where meaning relations are determined" (92). One of these interpretations, for example, is to determine scope: "The scope of A is the set of nodes that A c-commands in the LF representation" (133, emphasis added).

What bothers me the most about post hoc assignment of semantic representation is the entailment that I must be juggling words around, applying rules and making adjustments without knowing anything about what the thing is intended to mean. In fact, I am unable to find room for any semantic intention whatsoever in Chomsky. Is Transformational Grammar (TG) claiming that the rules up till S-Structure are totally devoid of any semantic considerations whatsoever? That is how extreme this position seems to be. How then did I decide what to pull from the lexicon in the first place? Is that random? If I pull out the lexeme house surely it is because I intend to refer to house. Why pull out terms I intend not to mention?

The other anomaly here is the modus operandi of interpretivism itself. In a near-Freudian denial, proponents of GB claim that theirs is not a production model. That stipulation is to excuse them from addressing the issue in the preceding paragraph. However, they fail to realize that a production model has two parts. As I move from Meaning to Form, (the part which TG patently dismisses as out of bounds) I, the speaker/encoder, choose a thought, then formulate and utter my message. At that point, it is received by the listener/decoder who does in fact have the task of interpreting the message. This is done by running the given message backwards through the rules. That is, (s)he takes the utterance and reconstructs what the underlying thought must have been. My point is that interpretation is necessarily a function in the production model. It is precisely half the production, and it is inside out. In order to surgically remove his GB portion of language, from the entirety of language production, as Chomsky is wont to do, he must explain along the way (or at least stipulate) how the relationship between these two allows for this removal. What is the point of having it survive on its own? Does it not eventually reconnect into the whole? To simply divorce a portion of grammar from the rest of human language production, without addressing such issues otherwise, makes generative grammar a naïve uninformed process. The point, then, is surely: what insights and advantages are there to this removal? I submit here that ipso facto it causes shortsightedness: we will have lost sight of the whole. Without the end in view, we more easily lose our way and even our destination.

What is the advantage of retaining production and using interpretation to run grammar in the other direction? The bias evident in doing so is well placed. Messages are necessarily biased to the encoder. If someone is confused with my message, or stumbles on an ambiguity, I have the sole right to expound my semantic intention, to interpret the message for the listener, because I encoded it, and if sober, know what I meant to say. It does little good for a listener to contradict me on that point, regardless of my
“performance” errors. The choice, then, is expressed in the question: Who better can determine for us the meaning of a sentence, the encoder or the decoder? If the answer is the decoder, then we should follow interpretivism. If the answer is the encoder, and I submit that it is, we then follow deterministic production.

This appropriate meaning bias of the encoder is reflected in starting with Meaning, or Logical Form. The irony of the label, using the term “form” in reference to a level of meaning, when in fact the dichotomy of form versus meaning is the most fundamental distinction in language, is further evidence of difficulty in their point of view: the stated purpose of Logical Form undoes itself in that it confuses form with meaning, and unwittingly becomes the second half of the very production model Chomsky has so vehemently and persistently eschewed since Syntactic Structures in 1956.

The Implication of a Terminal Checkpoint

Logical Form (LF) and Phonetic Form (PF) dovetail off of the syntactic component, because these have no input-output relationship to one another; no information is to pass between them. Since Late LF cannot feed PF, it is not possible to insert lexical items in Late LF and expect them to be “visible” to PF. This dovetail is not even handed, however. Flaws detected by either LF or PF can block a given derivation. So this dovetail must, in fact, reunite after the two components have run their course. For example, PF can continue finishing its work, while the sentence is blocked in LF. PF does not “know” this. Or vice versa, LF could pass on a sentence that does not make it through the checks in PF. I have to hear from both camps, before I know the sentence has “converged” as GB calls it. The checkpoint waits to hear from both wings of the dovetail before it can pronounce final judgment on the legitimacy of the sentence being derived.

The fact that there is no information passed between them is uneconomical, for while a flaw is found in one component, the derivation may be continuing successfully through the other component needlessly (since I now know that it will be rejected anyway), thus overgenerating and wasting effort. If I am interested in an LF judgment on an illegal sentence, I can run the derivation through, but that is an experimental procedure for theoretic value rather than the usual state of affairs of generating legal sentences through the grammar. Except for experiments, the earlier I stop misgenerating, the better off I am. Why bother finishing work in one tail, when the sentence has already crashed in the other?

If, on the other hand, we were to allow PF or LF to tell the other as soon as one of them crashed, we are right back where we started on the issue. If LF can tell PF it crashed, then why not let it tell PF that it just checked scope and, as we shall see below, may need to insert a marker for it? Any communication turns into all communication, and we lose the reason for separating LF and PF into incommunicado branches in the first place.

The PF-LF dovetail is a weakness in the program, based quite frankly on a stipulation that violates economy. Let us now discuss economy outright.

The Economy Principle

My guiding light is the economy and efficiency expressed in Occam’s Razor. For example, I want to constrain any excessive generative powers. It is wasteful to spend the energy of the grammar building structures for a late filter when an earlier one can intercept some of them and reduce spoilage. It is also uneconomical to disperse filters throughout the grammar if they can instead be consolidated. In that way, rules and checks are invoked less often while managing the same amount of control and functionality that was intended by the insightful generalities that resulted from the analysis.
Reducing *duplication of effort* is another economy. If a given semantic issue can be determined earlier in the derivation than waiting for Late LF, why maintain such a second component with those same abilities later on? Unless such a duplication of effort is justified (not stipulated) it should be removed to promote economy.

*Overgeneration* is another inefficiency. It is wasteful of the grammar to overgenerate structures with mismatched and illogical semantic markers in them, only to wait until near the end of the job (Late LF) to filter them out. If I have the information to exclude an illogical structure earlier in the grammar, it is more efficient to filter it out right then and there, because it reduces the amount and frequency of the overgenerated spoilage. An even better method would be to write the rule (at that point in the grammar) in such a way that a mismatch or illogical structure is never generated at all. Chomsky claims to support this principle:

Chomsky has suggested a condition on derivations called the “least effort” condition, understood as part of an overarching principle of economy of derivation. The interpretation of this condition . . . is that shorter derivations are always chosen over longer ones. This is to say that, in a situation where more than one derivation is possible for a given sentence, the one which involves [fewer] steps is chosen over the others. (Ouhalla 1994, 305)

But he must not see that it also applies in this case. In fact, one of the more amusing yet poignant pictures of excess in such an endeavor is the story of the room full of monkeys typing randomly away under the statistical charge that over an infinity of time they would produce a typescript of Shakespeare’s *Twelfth Night*. While it is possible for an English major to handle scrutiny of the output, it is always being *evaluated after the fact*. It is not heuristic; nothing drives the project towards its goal, such as using human typists could do. The project is not improving on itself as it continues.

As an aside, Optimality Theory could likewise learn a lesson from the monkeys. It, too, generates a potential infinitude of candidate forms in what has become linguistic history’s worst case of overgeneration to date (see McOmber 1994). To the extent that the checks-and-filters approach is also conducted after the fact and fails to provide intelligent cause-effect relationships in the output, such grammar behaves like the room full of monkeys. It behooves me then to prefer rule and principle driven mechanisms over randomized conundra. The latter is perhaps the ultimate expression of *ad hoc* at its very worst.

I have discussed economy as a guiding principle, even quoting Chomsky on the point, though I insist on applying it as broadly as possible. Let us now talk about the challenge to the integrity of LF that this creates, that is, the consistency of rules in it when considering the overall function and effect of the grammar.

**Modular Integrity**

*Modular integrity* in grammar compels components to be more self-contained than dispersed. If all meaning is consistently and only assigned in LF, I have a more consolidated, integral module than if only most of meaning is in LF. The latter would be a tremendously much more complicated stipulation: I would be forced to explain inconsistent boundaries—what criteria determine which types of meanings wait for LF and which ones come early? If *any* meaning is available early, then why not all? It would become my burden to explain that.

If scope is being determined and appealed to prior to the purview of LF, then the modular integrity of LF has been compromised. Still in all, when we find such breaches, it behooves us to admit it and repair the grammar. Far worse, and I have seen this repeatedly, is analysis where semantics is used, crucially, in the
explained phenomenon, and then
patently ignored as the rule is inserted
into the GB model. We will see several
cases of this below.

Chomsky’s insistence on avoiding the
production model appears to create more
work for all of us who are searching for
one. For if his grammar is not produc-
tion, then it is just another descriptive
grammar, a separate object from the pro-
duction version, and we suppose that we
need both: one to explain grammar in
general, and the other to explain perfor-
ance, both the flawed and the flawless.
On the other hand, if it turns out that a
production model of grammar does both,
as I submit it does, then we have killed
two birds with one stone and do not need
the other non-productive grammar at all:
Occam’s Razor eliminates it.

Production and production issues are
a reality. They are a part of language, in
fact, they are at the core: the purpose of
language is to communicate. I produce
an utterance and you have to guess what
I meant. Then, vice versa, you respond
and now it is my turn to try and infer
your meaning based on what you say, by
retracing your surface structure back
through rules to your original meaning,
as I hope I have correctly managed to
determine. Of course, further exchanges
should allow us both to clarify our com-
unication, and we learn the way each
of us expresses himself as though we
kept separate dictionaries on individual
nuance and style. We can and do.

Meaningful deterministic production
allows me to guide my grammatical
choices by knowing where I am headed. What
else, for example, guarantees that I
will ever generate a correct sentence if
there are no criteria determining the vari-
ous rule applications along the way? It
makes more sense to allow semantics to
steer the course rather than hoping we
happen to have all the pieces we need by
the time we get to a Late LF. If I build a
structure based on the point I want to get
across in my message, then I can guaran-
tee that the utterance has in it what I
need. That is responsible communication.
Choices in the derivation are based on
that guiding meaningful point I wish to
convey. Other choices may instead be
based on morphological, computational
or phonetic conditioning. But to deprive
entire derivations of ever having input
from semantics until they are finished, is
an unfounded stipulation without merit.
The randomization it requires is both
uninsightful and extremely wasteful.

It disturbs the integrity of a Late LF,
however, to make semantic choices while
generating sentences, since such knowl-
edge is not supposed to be available until
after the dovetail beyond S-Structure.
With this in mind, I now discuss the issue
of hindsight versus foresight which such
a dilemma inevitably raises.

Derivation: Hindsight versus
Foresight

I now submit, in good company from
other economical endeavors if not gram-
matical, that foresight is superior to hind-
sight in making choices. Determined gen-
eration is much more economical than
filters separating out needless spoilage.
Otherwise we relegate human speech to
the status of a computerized restaurant
where robots follow no specific recipe but
cook at will, randomly combining ingre-
dients, temperatures, and techniques.
Not to worry, we have tasters. Before any
of these concoctions is served, the maître
d’ takes a taste. He alone must either
serve the food, or else throw it in the
trash. But, crucially, he never talks to the
cooks! No feedback is allowed. He never
sends a soup back for more salt, or a rare
meat cut to be cooked till medium well as
the customer preferred. Never mind that
the customer is impatiently waiting for
his filet. He is not allowed to order any-
thing, even if it is his choice of entrée. He
just sits in the dining room hoping for
nourishment. There is, of course, no
guarantee that anything palatable ever
emerges from the kitchen. All the owner
of the establishment can promise is the probability that food of some sort will eventually come. No one can tell him when, what it might be, or how much will pass inspection: hindsight cuisine at its finest. In the meanwhile, the customer is welcome to peruse a Shakespeare manuscript from the nearby simian typing lab. It should be arriving any time now...

In the grammatical realm, GB is far too similar. There is a “sentence taster” at the terminal point where the dovetail covertly rejoins. The would-be filters provide no heuristic feedback to refine the grammar, to report what they have learned that tastes good. There are several judiciously early points in GB where available recipes could be followed and issues easily resolved by appealing to meaning. Recognizing a distribution of available meaning sites throughout the grammar is inconsistent with Late LF: “we will serve no meaning before its time.”

On the other hand, such procedures are evidence for Early LF since the semantics it provides persist throughout the derivation. I now present a crucial case where a GB analysis invokes semantics prior to Late LF, the overt scope marker *ma of Bani-Hassan Arabic.

**Semantics Prior to LF**

A semantic criterion prior to late LF is the overt scope marker: overt means pronounced in PF and therefore visible to it. But LF is deaf to PF, just as PF is blind to LF. If I insert it early enough for PF to have it, I am playing semantics out of bounds, and LF calls foul. If I wait till LF, PF won’t see it or pronounce it.

**Overt Scope Marking**

Bani-Hassan Arabic provides an example of the overt scope marker *ma. Kenstowicz (1989, 267ff, displays renumbered) explains that there is an LF rule . . . that helps to characterize the scope of expressions quantified by the negative existential *wolla. When an NP quantified by *wolla appears in postverbal position the verbs must be preceded by the particle *ma (2, 3), which in other contexts marks sentential negation (1). But

<table>
<thead>
<tr>
<th>Sentential Negation *ma</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) al taalib *ma gara al ktaab.</td>
</tr>
<tr>
<td>‘the student NEG read the book’</td>
</tr>
<tr>
<td><em>The student did not read the book.</em></td>
</tr>
</tbody>
</table>

Scope Marker *ma:

<table>
<thead>
<tr>
<th>Logical Form:</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>wolla ktaab; [al taalib ma gara e</em>]</td>
</tr>
</tbody>
</table>

(2) al taalib *ma gara walla ktaab.  |
‘the student NEG read no book’ |
*The student did not read any book.*

(3) *al taalib gara walla ktaab.  |
‘the student read no book’

(4) walla taalib gara al ktaab.  |
‘no student read the book’ |
*No student read the book.*

(5) *wolla taalib *ma gara al ktaab.  |
‘no student NEG read the book’

<table>
<thead>
<tr>
<th>Logical Form:</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>wolla taalib; [e</em> gara al ktaab]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logical Form:</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>wolla taalib; [e</em> ma gara al ktaab]</td>
</tr>
</tbody>
</table>
when the negatively quantified NP appears in preverbal subject position (4) the particle *ma* must be absent (5).

Kenstowicz determines that "*ma* acts as a scope marker indicating that the NP quantified by *walla* has scope over the verb" (268). In other words, *ktaab* can now have scope over *gara* (and hence the entire clause) despite its location lower in the tree. In sentence 4 the higher scope is already correct for *walla tialib* by virtue of its higher position in the tree, so no added *ma* is necessary, and is, in fact, crucially prohibited, as shown in sentence 5.

While the Kenstowicz material discusses the resultant semantics, it does not explain how the structure was built in the first place—how does *ma* get inserted, and when is the particular position determined? For if scope cannot be determined until Late LF, I will not know till then if I will need *ma*. Yet if it is determined that I will in fact need it, how can I insert it knowing it will then be invisible to PF? It would not be pronounceable. That is the paradox of an overt scope marker in Chomsky's system. Chomsky's way out is to randomly generate *ma* in the room full of monkeys and hope one of them types it in just the right spot. That is patently uninsightful, because it misses the generality that I do in fact know where and when I should insert it, at that point in the derivation. I am only waiting by stipulation. By refusing to address none but interpretive issues, and thus only the second half of production, Chomsky likewise misses insights and generalities associated with that first half.

The alternative is to determine scope prior to the dovetail. But if scope can be determined then, why redo it (duplication of effort) at LF? What does this second determination gain me? And is the earlier determination not an instance of an LF-like function (i.e., an earlier LF)? What is then the most efficient place in the derivation for scope determination? How can I eliminate a second (if needless) one?

Random generation of sentences with *ma* scattered throughout would be wasted overgeneration. It simplifies LF to not have to sort through all these if that can be avoided.

### Deriving the Walla Sentences

Let us walk through a derivation of sentences like 1–5 above. In the subcategorization of the V° *gara*, I choose a DP to be the patient/object/complement which builds a V', another DP, to be the agent/subject/specifier, which finishes the complex as a VP. Either of the DPs can be quantified by *walla*. Let us call a VP with *walla* in Spec, a High *Walla* not-ing that the *walla* is higher in the tree. VPs with a *walla* in their complement let us call Low *Walla* VPs. Referring back to sentences 2–5, High *Walla* structures have scope over the V' by virtue of having this higher position:

(6) Scope: The scope of A is the set of nodes that A c-commands in the LF representation. (Ouhalla 1994, 133)

This inherent or covert scope property explains why High *Walla* constructions will not need to have their scope marked overtly with *ma*.

At this point everything is fine, so far so good, but now the grammar reaches the *ma*-Insertion rule. Here is where the paradox unfolds. Notice first, however, that there is no GB step where some drastically new structure appears out of the blue. The various "set[s] of nodes" with all their c-commands which LF inherits were built step by step under full view. It is no surprise what structure LF gets, because we have just built it. In other words, as we are on our way to LF, we can see what is c-commanding what, and we realize what the implications will be. When we make a move that will have semantic implications later on down the derivation, why wait to fix it, or waste our time finishing a derivation which we know will crash, when we can built it properly in the first place?
I have shown you the scope rule and pointed out the advantage of perspicuity made possible by looking ahead (alias determinism). The reader is now, I believe, equipped to steer into the approaching storm and face the paradox head on.

The Approaching Storm

There are two possible ways to execute ma-Insertion. On the one hand, I can follow Chomsky and randomly insert ma without regard to the consequences (the consequences being the persistent spoilage of the overgenerated forms I know I will have to delete when I check for scope at Late LF). That approach would maintain a strict Late LF. The ad hoc pretense, however, is that I do not know what I am saying—that I cannot intend a particular scope when I wish to—and then guide the derivation accordingly. I cannot look ahead.

Why am I inserting ma? In Chomsky's approach, the grammar does not care. Not only does it not care, it cannot guarantee that the insertion will ever take place. There is only a theoretical probability that the ma rule will ever be chosen. Nothing, in fact, beyond the infinitude of time and sample size, ever guarantees in GB that I ever will generate a ma. Instead of choosing wisely and deliberately, I am asked to flip a (hopefully fair) coin. I have to gamble for grammar!

The other choice, iconoclastic to GB, is to check deliberately for scope now, while I am on my way. This is what I call determinism—the determinism of a production model. I have the information at that point of the derivation, to determine the correct choice, so I will use it. There will be no need to filter out poor versions, because I won't choose any.

The point would be that I "know enough" to do so (even though such knowledge should be relegated to Late Logical Form). The upshot, of course, is that I even knew at the beginning of the derivation. That is precisely what I mean by a production model. Here speakers know the thought they wish to convey, and have it in mind as they formulate the sentence.

I now consider the consequences of each of these two, in turn.

Alternative One: Random Insertion

If I follow the first course and randomly insert ma, the only guarantee that I will ever generate a sentence correctly comes from the infinitude of statistical probabilities of sheer chance. This technique is guilty of overgeneration since it will just as likely produce as many incorrect sentences as correct, on a statistical average. That is the coin toss effect—heads: we insert ma, tails: we don't. Yes, it is statistically possible, albeit rarer with increasing n, but there is a 1 in 2^n chance of generating the sequence of heads and tails you chose. There are then only four combinations I can generate. These are shown in 7–10 below, with the corresponding examples repeated from sentences 2–5.

Only 8 and 9 are legal. In 8 since the walla is high, it can function as the scope marker and no walla is needed, in fact, it is strictly disallowed in this reading.2 In 9 walla needs to have scope over V' and thus a Low Walla construction must be marked overtly with ma.

This review completes the Kenstowicz (1989) material that is crucial for the discussion here.3 I claim that it is inappropriate to allow grammars to ignorantly apply a rule in the face of a criterion that is available and could have been applied. This is a heavy price to pay for the stipulation that LF be postponed. Stipulations should be reserved for plugging loopholes until further insight can be found.

We have seen random insertion, which produces at least as many bad structures as it does good ones. The other alternative is deterministic insertion.
Possible Combinations

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(7)</td>
<td>High *walla / with *ma</td>
<td>Examples</td>
</tr>
<tr>
<td>(8)</td>
<td>High *walla / without *ma</td>
<td>*walla ṭaalib *ma gara al kitaab. = (5)</td>
</tr>
<tr>
<td>(9)</td>
<td>Low *walla / with *ma</td>
<td>walla ṭaalib gara al kitaab. = (4)</td>
</tr>
<tr>
<td>(10)</td>
<td>Low *walla / without *ma</td>
<td>al ṭaalib *ma gara *walla kitaab = (2)</td>
</tr>
</tbody>
</table>

Alternative Two: Deterministic Insertion

The overgeneration hypothesis allows us to generate all four sentence types, 7-10, and then filter out the wrong ones at Late LF. This rule for inserting *ma in such a grammar is also uninsightful. It optionally inserts *ma to the left of the verb (e.g., in an adjunction to V') without a motive.

If, however, I supply the motive, which is to mark scope, then I am admitting that I can determine scope at such an early point. That would be the informed approach. Then the *ma-Insertion Rule “knows what it is doing” and only generates 8 and 9 without generating 7 and 10 because the rule does not apply. Such insight has an impact on economy of derivation. For if I can and do “determine scope” at such a point prior to S-Structure, I have two ways to look at what has happened. On the one hand, I am, according to definition 6, to determine scope at LF; so LF must have come earlier than expected. On the other hand, Late LF is going to check to see that I did the scope determination correctly when I get down to the dovetail. If, as option a, I determined scope correctly, the check is a waste and that part of LF is useless. If on the other hand, as option b, I determined scope incorrectly, the *ma-Insertion rule itself is a waste as written and needs to be fixed. But that is also not a permanent contribution of a Late LF. Once the rule is fixed, I am back to option a for a useless Late LF. Notice, moreover, that I cannot let Late LF repair a faulty 10 by supplying the missing *ma, because invisible PF can never know about the change and will mispronounce the output. Likewise, Late LF cannot repair a faulty 7 by deleting an extraneous *ma for the same reason.

I have presented the data and analysis and explained the consequences to the model. I will now summarize these into a restatement of the paradox.

The Paradox

The paradox in GB’s dovetail is this: if I mark scope with *ma in LF, I cannot hear *ma in PF. If I hear *ma in PF, it must have been inserted prior to LF, and therefore prior to the dovetail at S-Structure. That would mean that a meaningful semantic-based act occurred prior to the semantic component. The data show that the marker is phonetic yet also semantic. How can it be both? The one contradicts the other in GB.

This paradox in the theory hinges on the principle of economy of derivation. If scope is determined only at Late LF, we have an uninsightful and overgenerating insertion rule. It is uninsightful because it inserts randomly, ignoring the very criterion which will be considered in a module that is placed later by stipulation. It overgenerates because the probability of the random event with two outcomes is 1/2, and even that assumes that structures without the scope marker occur as frequently as those with it.

If scope is determinable earlier than S-Structure, providing insight to *ma-Insertion would thereby reduce the overgeneration. It eliminates the need for a Late LF filter and eventually depletes the reason for having Late LF at all. For Late LF only duplicates checking processes that have gone on before, I will want to eliminate these by the same principle of economy as Chomsky invokes for his Minimalism.

All this conflict is resolved by Early LF, which captures the best of both worlds. Early LF recognizes the Meaning
level which drives many choices. It allows the ma-Insertion to be insightful. It precludes overgenerating 7 and 10.

The simplest way out of the paradox is to move the scope function to the earlier position. Let us now look more closely at the Early LF solution and its economies to the grammar.

**The Solution: Early LF**

To look at the economy of an Early LF solution, let us look at the trade-off in eliminating the filter that was checking for illegal occurrences of ma.

In the Late LF model, the correct example in 2 above, repeated here as 11, successfully passes a filter:

(11) \( \text{walla ktaab} i \ [\text{al \ t\aaalib ma gara e}_i] \)

This LF shows the quantified ktaab in scope over a clause in which it is otherwise subordinate. In such a case, the filter checks for \( ma \) to the left of the verb. Let us call this the ma-Scope Filter, and the principle behind this filter the ma-Scope Principle. Thus 11 is deemed to be a legal sentence. However, the putative LF for the incorrect 3, \( \ast \text{al \ t\aaalib gara walla ktaab} \), does not pass the filter for lack of \( ma \):

(12) \( \ast \text{walla ktaab} i \ [\text{al \ t\aaalib gara e}_i] \)

Rather than waste the derivation, however, why not let the very ma-Principle which provides the filter provide a remedial rule instead? Rather than having the principle create a filter, have it create a rule, the ma-Scope Rule. Then, rather than throwing 7 away, I repair it by correctly inserting the \( ma \) to the left of the verb:

In an Early LF grammar, rather than Quantifier Raising, I have its mirror image, Quantifier Lowering, which replaces the bound variable \( e_i \) with its quantifier \( \text{walla ktaab} i \):

This gives us the correct 2. I have just traded a filter for a rule. In so doing, I avoid overgenerating 3 which gets filtered out. That adds economy by reducing wasted by-product. The ma-Scope Rule indirectly "checks" for 3 by deriving it into correct 2. In fact, that is the only way 2 is derived. There is no reason to have \( ma \) in the sentence until the ma-Scope Principle puts it there. It explains both why (the scope) and how (the insertion) \( ma \) is found in that position. Since this principle is semantic, however, it belongs in an LF module. When LF is early, it can insert items and they can persist to PF.

In the Late LF model, I am left with the inconsistency that a scope marker inexplicably may be inserted in a previous component of the grammar but not be checked until I have spent energy deriving the entire S-Structure. If the \( ma \) was not inserted according to a proper analysis of scope, just how was the insertion decided? Could it also have been misplaced as well, say, as in 14?

(14) \( \ast \text{al \ t\aaalib gara ma walla ktaab} \)

Why should the grammar allow 14 to persist into later components and undergo anymore scrutiny or derivation?

If LF cannot feed PF, it should not be possible to insert lexical items in LF, for they will be phonetically invisible. Yet if it is in LF that scope is determined, it is only at that point (or one subsequent to it) that such a marker can in fact be knowledgeably placed. What determines the placement otherwise? If placement is random, I am overworking the filter, because the grammar by chance will wastefully generate many more incorrect structures (which will have to be weeded out) than correct ones. In a program with

---

Derivation of (2) al ātalib ma gara walla ktaab:

(a) Given: \( \text{walla ktaab} i \ [\text{al \ t\aaalib gara e}_i] \)
(b) Apply ma-Scope: \( \text{walla ktaab} i \ [\text{al \ t\aaalib ma gara e}_i] \)
(c) Quantifier Lowering: \( e_i \ [\text{al \ t\aaalib ma gara [walla ktaab} i ]] \)
Late LF, the marker will have to be inserted prior to the dovetail in order for PF to pick it up and spell it out phonetically. That is the paradox of making semantic decisions prior to the semantic component itself. I am faced either with loss of modular integrity or with overgeneration.

There is no loss of economy by moving the filter earlier and turning it into a rule. If a scope marker inserted for semantic reasons, it must be in LF. If it is heard, it must insert prior to PF. Therefore, LF must occur prior to PF. Quantifier Lowering can provide D- and S-Structures so that LF does not need to follow these. Therefore, it can be the earliest module of all. That is the escape from the paradox. Let us now look at other consequences of an Early LF program.

All of the problems above are simplified or solved when the LF component is placed first. That is, if the derivation runs from Meaning to Form as the utterance is composed, then the decoder’s task is to decompose this from the given Form back again to its intended Meaning. I have pointed this out as I went. Here now are some additional consequences of Early LF.

Semantic Selection and Theta-Roles

An additional consequence of Early LF concerns another semantic pretense, similar to the scope case above. This time I address the semantic selection and theta-roles, a very early function and hence an interesting one if LF is going to narrowly precede it. Notice the appeals to meaning and even determination in the following explanation:

S-selection is largely determined by the inherent meaning (the conceptual structure) of lexical items. The verb (or concept) hit, for example, entails two participants ... s-selection operates in terms of semantic categories, called thematic roles or theta-roles. (Ouhalla 1994, 125 emphasis added)

It violates modular integrity to claim that the Theta Criterion “applies at LF . . . but extend[s] to other levels of syntactic representation by virtue of the Projection Principle” (Ouhalla 1994, 126), because I am then also forced to conclude also that LF is “essentially a syntactic level of representation” (126). Late LF looks more syntactic than necessary because it must take S-Structure as input. By contrast, Early LF is relieved of all but the most transparent S-Structures because the rules from S-Structure have not even applied yet. Therefore, Early LF is less syntactic, overall, than Late LF. That promotes economy.

Logical Subjects and Objects

Reference to logical subjects and objects also belongs to LF, since logic is a part of semantics. An appeal is made to these concepts, for example, to explain the DP movement in long passives—“the term ‘subject’ is used here in both its structural sense . . . and its semantic sense” (Ouhalla 1994, 78f emphasis added). Then why position LF so far down the derivational track from yet another fairly early point (as early as D-Structure), when in fact it would serve to have it available as soon as the first mention of semantic is made? GB’s answer is supposed to be that such is the exclusive domain of LF and that (per modular integrity) this shall not be violated.

While such persistence seems to promote consistency at this point, I now turn to an issue where it does not fair well at all: the issue of reconstruction effects.

Reconstruction Effects

Late LF structures have a lot of S-Structure material in them that is not relevant to a determination of meaning and do not enter into LF considerations. By its very nature, Late LF first inherits S-Structure as is, and is then allowed only covert movements to get the job done of determining meaning. But much of the idiosyncracies of language have already
been encoded, so to speak, into the various adjustments made since D-Structure. Early LF would not burdened with these, because they result from rules that would not have applied yet.

Chomsky (1995, 71 display renumbered) debates whether to allow reconstruction effects, or to admit that LF is not doing it job alone:

Some semantic properties . . . appear to be determined by S-Structure . . . independently . . . of the LF component. Let P be such a property. Then two accounts are possible.

(15) a. P holds at S-Structure

b. P holds at LF under reconstruction, that is, with the moved phrase treated “as if” it were in the position of its trace.

In fact, either alternative is ultimately fatal for Late LF: 15a patently so, for if S-Structure is allowed to do some of the semantic work, then why not D-Structure also, in which case, let D-Structure do all of it to avoid loss of modular integrity and turn it into Early LF with all its efficiencies. On the other hand, if we opt for reconstruction, then we will need the inefficiency of additional mechanisms to support it.

Reconstruction involves structures with a moved element, NP for instance, where structural relationships (linear precedence and c-command) are crucial to determine the antecedents of various pronouns (see Chomsky 1995, 73f).

In 16a he c-commands John and therefore cannot take John as its antecedent. That is expected. Likewise in 16b he does not c-command who or its trace e so that he could in fact refer to the same person, or not. The lack of c-command prevents the restriction against co-reference. There are two meanings possible, 17a and b.

The reconstruction effect comes in with 16c. In that sentence, whose is not c-commanded by he either, and yet Chomsky does not want it taken as its antecedent. His point is then that whose father acts as if it is in the original (premoved) position, because then he clearly c-commands whose and is supposed to preclude co-reference.

Notice, however, that if whose father has been moved to the left for S-Structure, and we are having to move it back for LF to interpret it, that we could have interpreted it back before it moved in the first place. In other words, this is another duplication of effort violation. Moving the NP back to its trace, duplicates the earlier level where the NP was. That first position is therefore meaningful, and the moved position is not. Refusing to use the meaningfulness of that first position is not principled, but only a stipulation of GB. Notice that if 16b is reconstructed into 19, no ill effects

<table>
<thead>
<tr>
<th></th>
<th>(16)</th>
<th>(17)</th>
<th>(18)</th>
<th>(19)</th>
<th>(20)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td>They said he admires John’s father.</td>
<td>b.</td>
<td>Someonei said he admires John’s father, who0 is it?</td>
<td>Guess [whose father]i [they said he admires e0].</td>
<td>He ≠ John</td>
</tr>
<tr>
<td>b.</td>
<td>[e0 said he admires John’s father]?</td>
<td>c.</td>
<td>Guess [whose father]i [they said he admires e0].</td>
<td>He ≠ whose</td>
<td></td>
</tr>
<tr>
<td>c.</td>
<td>Guess [whose father]i [they said he admires e0].</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(19)</td>
<td>[Who] said he admires John’s father]?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(20)</td>
<td>Guess [whose father]i [they said he admires [whose father]i].</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
result for interpretation 19 because the c-command relations were not disturbed.

The so-called copying solution to reconstruction is even less promising. The item is not actually moved in this have-your-cake-and-eat-it-too approach, but rather copied to its new location. This preserves original c-command relationships. Presumably only the copy is discarded in LF, and only the original is discarded in PF. But that is also why it is inefficient: the two NPs look in 20 like they are on equal footing, yet the grammar, as we have just stated, will treat them with bias just like before when it gets to the dovetail, with PF taking the copy, and LF the erstwhile trace (original). It is still an instance of preserving the earlier structure in some way (as in Early LF) in order to use that structure later to help with interpretation.

Reconstruction raises the issue of whether Late LF is really doing its job. Another indication that it is not, after all is said and done, comes from Chomsky himself, and deserves brief mention.

Defective Convergence

Another advantage of Early LF is that it allows me to eliminate a level of representation. D-Structure now becomes an arbitrary point on the way from Early LF to S-Structure. The notion that D-Structure is the level "where all categories are in the positions where they are expected to be" (Ouhalla 1994, 56) and that S-Structure moves or chains them away is circular: they are where we expect, because we stipulate our expectations into the rules as we write them. If we can then apply said rules accurately, we can hardly be surprised if the outputs are now where we expect.

Where does that leave the Arabic ma particle? How do I decide where to expect ma to be in D-Structure if I cannot determine requisite scope until a Late LF? But notice that since “expected positions” refers to the accurate output of previous rules, Early LF places things where they belong sooner in the derivation. There is no arbitrary point along the way from Early LF to S-Structure to claim as an independent D-Structure level. S-Structure rules can take us directly to the surface. D-Structure is thus extraneous.

Parasitic Gaps

Parasitic gaps are licensed at S-Structure by a variable which does not c-command it, a condition “sometimes called the anti-c-command condition” (Ouhalla 1994, 218, notation adjusted). However, we are told that this licensure is not to apply beyond S-Structure into LF. Variables which arise from QR at Late LF are unable to license such a gap:

(21) S-structure:
     *I forgot who filed every/which article without reading e₁
Late LF:

*every /which article [I forgot
who filed ei [without reading [pi]]]

where pi is parasitic on ei. Ouhalla (1994, 219) wisely comments that the parasitic
gap licensure restriction “should be
derivable from some general and inde­
pendent principles, although it is not
easy to see how” given Chomsky’s
framework. With Early LF, however, the
licensure can be invoked at the appropri­
apate point in the derivation and stay valid
through to the end, without backwashing
into the earlier invalid levels of Meaning.
Early LF never interferes with subse­
quent licensure authorities of S-Structure,
because S-Structure follows rather than
precedes Early (but not Late) LF.

I have presented several additional
points that Early LF resolves economical­
ly. Another issue affecting economy is
procrastination of rule application.

Procrastinate versus Expedite

Procrastination is a Chomskyan
(1995, 228) principle that delays doing
something until the last possible point. I
prefer to expedite functions in order to
build the most efficient grammar (MEG)
possible (McOmber 1979). Switching
from a principle of procrastinate to expedite
supports moving LF early as well.
Instead of waiting until LF (as in English)
to check features because they are strong
(see Cook and Newson 1996) the features
will have been used to drive the correct
morphology in the first place, thus more
cost avoidance for the grammar.

Chomsky’s Greed principle follows a
similar fate. Instead of node hopping to
feature check, I treat all agreement as
post-semantic feature copying (see Chafe
1970, ch. 5ff). After choosing “the tasty
tacos” from the Spanish lexicon I copy
masculine and plural information from
“taco-s” over to “tasty” and the.” I
change underlying l taco-s sabroso to l-o-s
taco-s sabroso-s. Why bother generating el,
las, la and then “check” to eliminate the
wrong ones after the fact, when the infor­
mation is at my finger tips? I have
already decided to say taco-s as a plural
since I determined there are more than
one of them. How far away from produc­
tion dare we move—and what does it
accomplish? Whatever other abstractions
we achieve, we must eventually deal
with production to explain how language
communicates. And why not start with
this end in mind?

Tasty tacos is an example of deliber­
ate feature copying versus post hoc fea­
ture checking. Now let us look at a longer
distance phenomenon, (dis)anaphora.

Disanaphora

Other examples of scope that are
overtly marked in PF, are the intonational
variances we use to create what is tradi­
tionally labeled contrastive emphasis, as
in these examples from McOmber (1977
and 1978):

(22) a. The boys who are póór need money,
    but the others don’t. {B ∩ ~P]

    b. The bôys who are poor need money,
    but the others don’t. {~B ∩ P]

    c. The bôys who are póór need money,
    but the others don’t. ~{B ∩ P]

The emphatic intonation on póór or
on bôys or both, each leads to a respective
difference in meaning for the reference of
the complementary disanaphor others:

(23) a. [B ∩ ~P] means that
    others = boys (but not any girls)

    b. [~B ∩ P] means that
    others = poor girls (no boys)

    c. ~(B ∩ P) means that
    others = rich boys and poor girls

The phonetic signals necessary to
interpret data in 22 is heard in PF, but
Late LF is deaf to these. Here are further
examples based on Williams (1997, 605):
(24) a. *Sue wore a blue gown knowing Jill would wear a red one.

b. Sue wore a blue gown, knowing Jill would wear a red one.

In 24, one is anaphoric to gown, while blue is disanaphoric to red, in deliberate contrast. This pairing sets up a contrastive emphasis in the intonation that is used even for numeric data (619 mutatis mutandis):

(25) a. “...then dial extension 8 4 3 4.”
   (eight four three four)

b. “...then dial extension "8 4 3 4."
   (eight four three four)

As an aside, Williams finally concludes that LF = PF. In other words, for him there should be no dovetail. Such a move would overcome the disability of LF to hear crucial phonetic clues, but would not solve the blind overgeneration problems discussed above, nor the resulting problems of semiproduction GB incurs when we generate structures randomly, ignoring available semantic determinants. Therefore, while I agree that there is no dovetail, the fact that Late LF = PF is even clearer from a production model: the PF heard is actually the beginning point of interpretation (the erstwhile LF); that is, on the hearer’s track tracing and analyzing Form back into Meaning.

Zero Syntax

I modify Zero Syntax (pace Pesetsky 1995) to show that structures fresh out of Deep Meaning look like his layered structures, and then are linearized (à la Chafre) into the cascade-like trees as we proceed toward Surface Form. I am thus resurrecting Groat (1992) in a crucially mirrored image to obviate the fatal flaw Pesetsky found. I also depart from Groat with my Early LF.

Summary of Evidence for Early LF

I have presented several examples of issues that promote simplicity and economy in an Early LF. After explaining the Arabic overt scope marker, I also took a look at the concept of “logical” subjects and objects, the irony of reconstruction effects, the extraneous of D-Structure, the struggle of parasitic gaps, then I contrasted the principle of GB’s procrastination with MEG’s Expedite—all in the name of economy to the grammar. Finally, an additional example of clash between PF and LF shows up with the contrastive emphasis of disanaphora.

In every case, it is possible for GB to obviate the (efficient) position of Early Meaning by overgenerating and filtering out. Even if such filters appear equivalently accurate, and I do not argue that moot issue, the determinative Early LF model is, ceteris paribus, a more highly valued grammar, judging by Occam’s Razor.

Having reviewed evidence for Early LF, I now recapitulate my conclusions.

CONCLUSIONS

The scope of this paper is to argue on Early Logical Form. As I continue to repair the GB model, at the risk of pouring “new wine into old bottles,” I see that after moving LF, there is no longer a clean point which I can distinctly call D-Structure. It is, in a sense, absorbed by LF (I could use either label). Then there is little motivation for calling the midpoint “S-Structure,” in particular, since it is just as much surface of the meaning component as it is the deepest level of final form. New labels would be more insightful and all the more conveniently abbreviated.

If I have Early Logical Form, this is the meaning end of the processing. The encoder has a meaning in mind and wants to express this into a surface Form. The gamut then runs from Early or Deep
Meaning, through all the semantic and postsemantic processing (à la Chafe 1970). Then I arrive at the Late(st) or “Surface” Meaning. That comprises the boundary between Meaning and Form. Late Meaning is Early Form; that is, the end of the first module is the very starting point of the second. Let us simply call it the Midpoint, with that understanding that it is the “Checkpoint Charlie” between the two. Then from Early Form I have various phonological processes that take me finally to the surface: Late (Surface) Form. That is how I encode a sentence. The listener, to decode, runs backward through the derivation to arrive at the early “deep” meaning which was conveyed. The decoder must retrace steps, disambiguate, and in a real sense undo the derivation as it was encoded in the first place, running back again, from Form to Meaning.

Let me reiterate that I am following the economy principle of Occam’s Razor which leads me to (1) avoid duplication of effort, (2) avoid overgeneration with its spoilage and extraneous filters, (3) promote modular integrity, which tells me to keep LF intact, rather than splitting it between scope insertions and scope filters, (4) promote perspicuity, changing hindsight to foresight wherever possible, to add insight and oversight to the various processes in lieu of random ones that ignore available criteria, (5) equate Late Meaning with Early Form, and (6) use a processing model that follows the flow from speaker to listener (encoder to decoder) or back again, by running forward or backward respectively through the paradigm.

The undoing of Late LF model was based on the patent impossibility of having an overt scope marker, computed in LF but paradoxically invisible to pronunciation at PF. Moving LF to the front of the derivation precludes the paradox but changes the scope (pun intended) of LF considerably, from an interpretive to a deterministic production model. The code then runs from Meaning to Form and back again.

My conclusion is to abandon all interpretivistic models in favor of the most efficient grammar possible: a bilateral production model. It first encodes a sentence from Meaning to Form: that is expression. It then reverses the process back again to decode, from (the given) Form back to Meaning: that is interpretation. It is ironic, then, to compare this picture with Chomsky’s interpretivism and realize that his is actually a partial production model. Interpretation is always a decoding process. That is, it is the search for the original meaning of the message, that is lurking somewhere behind the form. Interpretation takes place on an utterance that was built with some original meaning in mind as the driving force generating the structure. Chomskyan interpretation is possible, at the end of his derivation, but he crucially leaves out of the derivation the original occurrence of the very interpretation that the encoder used in the first place, and for which the decoder now seeks. How much better for the grammar to run through the rules, produce an interpretation, and then match this with the original as a proof. For grammar to otherwise become so abstract that it manipulates structures while abstaining from recognition of the meaning that underlies those forms, is to take from grammar its very raison d’être: that grammar is the tool used to express thought into language.

NOTES

1. Pluralitas non est ponenda sine necessitate, roughly “necessity is the mother of complexity”—William of Occam, ca. 1285–1349, a Franciscan minimalist, probably did not invent the phrase, but quoted it so effectively and so often that it has since been attributed to him honoris causa. Following St. Francis all the more closely, he struggled with the Pope (John XXII) over various issues, who in turn excommunicated him. True to form, William responded with a treatise showing that the logic used in his own excommunication entailed that the Pope was a heretic himself.
2. The other possible reading, [there is] no student who did not read the book, uses sentential *ma* which is then the standard negative. The double negatives cancel, meaning that all the students read the book.

3. His data were given to discuss extraction with the null subject parameter. My focus with his data, however, is on how and where the *ma* comes to be inserted itself.

4. Again, this is according to Chomsky: hoist with his own petard. I see a possible answer where he does in fact refer to whose: "You'll never guess whose father he admires: his own!" Our point here is not to explain the structure our way, but to use Chomsky's own accepted evidence to refute himself.

5. To make matters worse, instead of just saying that the moved piece was copied into its new place, the literature still claims to the moved NP was completely moved away (deleted from its original position), and that the original was then restored by having copied the moved version back. That spends (wastes) twice the effort. Let the reader imagine word processing that way.

6. In fact, government itself (the G of GB) is also at risk. Last reports are that it can no longer remain "as a fundamental notion of the theory" (Cook and Newson 1996, 316).

**REFERENCES**


