Á. Gallego & D. Ott (eds.). 2015. *50 Years Later: Reflections on Chomsky's* Aspects. Cambridge, MA: MITWPL.
Available for purchase at http://mitwpl.mit.edu
© The Author

GENERALIZED TRANSFORMATIONS REANIMATED

EDWIN WILLIAMS
Princeton University

Chapter 3 of *Aspects* (especially pp. 131-137) presents two arguments for moving the "recursive property" of language from the transformational component to the base component. Although the arguments were compelling in the context of the findings in 1965, they are no longer, and in fact I think that something like the "generalized embedding transformations" of the pre-*Aspects* model may indeed be an essential part of an explanatory theory of syntax.

Prior to *Aspects*, the phrase structure rules of the base generated finitely many simple clauses; the "recursive property" arose from the operation of "generalized embedding transformations", which embedded clauses from the base inside one another to build recursive structures[1]. So, for example, from { [John said S'], [he was sick]}[2] one could get {[John said he was sick]} by applying a generalized embedding transformation, which substitutes a clause from the base for the dummy marker S'. The diagram of embeddings in a derivation was called a "Transformation-marker", by analogy with "Phrase-marker". The other transformations, called "singulary transformations," operated on the simple clauses of the base to derive their surface forms.

In the following passage, Chomsky observes that this arrangement permits derivations for which there is no apparent need:

> There are no known cases of ordering among generalized embedding transformations although such ordering is permitted by the theory of Transformation-markers. Furthermore, there are no really convincing cases of singulary transformations that must apply to a matrix sentence before a sentence transform is embedded in it, though this too is a possibility, according to the theory. (p. 133)

[1] The fact that the base had NP/PP recursion was ignored in *Aspects* and will be ignored here also; but see Williams 1998 and Williams 2003 for considerations that suggest that clausal embedding is fundamentally different from NP embedding.

[2] Braces enclose the "Workspace" in modern terms; in the pre-Aspects model, one might think of the base as the union of all Workspaces.

In order to eliminate these apparently unused possibilities of the theory, Chomsky proposes the model that we live with today:

> These observations suggest a possible simplification of the theory of transformational grammar. Suppose that we eliminate the notions "generalized transformation" and "Transformation-marker" altogether. In the rewriting rules of the base (in fact, in its categorial component) the string #S# is introduced in the positions where in the illustrative example we introduced the symbol S'. That is. wherever a base Phrase-marker contains a position in which a sentence transform is to be introduced, we fill this position with the string #S#, which initiates derivations. We now allow the rules of the base to apply cyclically. preserving their linear order. (p. 135)

A unified base that encompassed all recursion, combined with cyclic application of the singulary transformational rules (p. 134), eliminates much of the indeterminacies of rule-interaction that existed before. Doing clausal embedding in the base fixes the interaction of clausal embedding and singulary transformations, and the transformational cycle fixes the order of singulary transformations applying in different clauses. Within a single clause, Chomsky observes that the singulary transformations might be partially ordered, whether by universal stipulation, or on a language-particular basis.

As a further argument against generalized embedding transformations, Chomsky suggests that the theory with a unified base makes semantic interpretation easier, as interpretation does not need to consult the Transformation-marker, as all relevant information about what is embedded in what is now represented in the base.

In order to regiment the ordering of singulary transformations, Williams 1974 proposed a "Supercyclic" theory. The cyclic theory suggested in *Aspects* says that no rule applies to an embedded clause once any rule has applied to the embedding clause. A Supercyclic theory says that no rule applies to an embedded phrase once any rule has applied to the phrase that embeds it. Supercyclicity is simply cyclicity with more granularity; in fact, with the most granularity. In the 1974 model there were four phrases that formed the "spine" of the clause: VP, PredP, S (= modern TP), and S' (= modern CP), and rules operating in different ones of these domains within the same clause were ordered with respect to each other; it remained, though, that rules operating within the same domain were not ordered with respect to each other by the theory. Supercyclicity thus removed some, but not all, of the looseness about ordering in the *Aspects* theory.

In the intervening years, the structure of the clause has enriched enormously. If we accept "cartographic" findings, instead of 4 layers of structure in a clause, there are over 100, and those ordered elements compose the "F-structure" of the clause. The result is that it is virtually guaranteed that Supercyclicity eliminates all of the play in the ordering of the singulary transformations in the *Aspects* model. If it looks like two different operations target the Spec of the same F_i, where F_i is an element of F-structure, then there is probably an F_{i+1} just above F_i that one of the rules targets instead. If this thinking is carried out thoroughly, there will be a one-to-one relation between F-structure and operations, and thus F-structure fully orders operations under Supercyclicity. With clausal recursion in the base and with Supercyclicity, all of the looseness in the ordering of operations is eliminated.

The story could stop there, but in fact Supercyclicity combined with rich F-structure offers a new opportunity to compare the two models of clausal embedding that are compared in *Aspects*. In *Aspects*, the choice was between "in the base" and "in the transformational component." In the modern context, where the base and the transformational component are not separated, the choice must be reformulated. A first attempt to resurrect the question might be put as a choice between these two:

(1) a. Clauses are Merged with the verbs they are complement to[3]
b. One clause is embedded in another clause after both are formed

(a) is the "Base Theory" (BT) of clausal embedding in the modern context, the one that stems from the decision made in Aspects; (b) is the Modern Generalized Embedding Theory (MGET), the one that stems from the system that chapter 3 of *Aspects* puts aside.

A typical derivation of each is illustrated here[4]:

(2) a. BT: { think_{v}, $[\text{that Bill left}]_{CP}$ } → { $[\text{think [that Bill left]}]_{VP}$ }
b. MGET: { $[\text{Mary } [\text{think}]_{VP}]_{CP}$, $[\text{that Bill left}]_{CP}$ } →
{ $[\text{Mary } [\text{think } [\text{that Bill left}]_{CP}]_{VP}]_{CP}$ }

BT, like the *Aspects* model it descends from, is immune to the arguments in *Aspects* against generalized embedding transformations, since it doesn't have them. MGET, on the other hand, raises again the questions about how the embedding operations are ordered with respect to each other and to other operations[5], and so on those grounds BT is to be preferred, and it is in fact the modern default standard.

However, the high granularity of F-structure offers an opportunity to adapt MGET in a way that nullifies the *Aspects* chapter 3 arguments. The richness of F-structure allows for a variety of clause types, where types are distinguished by "size" in F-structure terms. For example, suppose that 4 distinct points in F-Structure correspond to VP, vP, TP, and CP (in that order; that is, VP<vP<TP<CP). These define 4 sizes of clause, where, under the simplest assumptions, each is independently embeddable. vP might be the size of the complements of *see*, as in "I saw John move"; TP complements might be the size of complements to *seem* that allow raising to subject; and CP might be the size of full tensed indicative *that*-complements. Verbs will be specified for the size of their complement clauses: *see* for vP, *seem* for TP, *think* for CP. And given that F-structure is rich, with more than 100 points, there could be that many further clause sizes.

F-structure thus regiments three separate things: the structure of clauses, the order of operations, and the size of embeddable clauses. We are now in a position to answer Chomsky's question about the theory MGET: how does clausal embedding interact with other operations? The more particular question to be answered is, when does clausal embedding occur? And since F-structure orders other operations, the pertinent question is, when, in F-structure terms, does clausal embedding take place?

The answer given in Williams 2003 was, "when the clause is big enough." This answer makes sense in both BT and MGET. In BT it is trivial; if *see* takes a vP complement, we will build up the vP, and then merge it with *see* to make a VP:

(3) {see, $[\text{Bill run}]_{vP}$ } → { $[\text{see } [\text{Bill run}]_{vP}]_{VP}$ }

---

[3] For simplicity only complement clauses are considered here.
[4] Braces enclose the "Workspace"; commas separate items in the Workspace.
[5] In the modern theory with "internal" and "external" Merge, there is not the separation between "the base" and "singulary" transformations, so I have dropped those terms in favor of simply "operations."

What is perhaps less obvious is that the answer also makes sense in MGET. In that theory, all clauses are generated as separate elements in the Workspace, and then are embedded in one another, as in the pre-*Aspects* model. The notion of "when" must be cashed out in terms of F-structure if we are to tie together the behavior of embedding operations and other operations, for, as we have determined already, the timing of other operations is fixed by F-structure and Supercyclicity. So the timing of clausal embedding has to be fixed by F-structure, but how?

As F-structure governs the structure of the clause, the size of embeddable clauses, and the timing of movements, it seems fitting that F-structure be thought of as more abstract than any one of these things. We might think of F-structure as a kind of clock that timed various operations. It is a clock which starts at F_0, and advances to F_1, then to F_2, and so on to F_n, and a derivation consists of a single sweep of the hand. Every action in the workspace is timed by this clock. As a consequence, all the clauses in the workspace "grow" simultaneously. In terms of the four F-structure points we have singled out, VP, vP, TP, and CP, this means that all clauses start out as VPs, then are built up simultaneously to vPs, then to TPs, and then to CPs.

With this assumption, it is perfectly clear what it means to say that a clause is embedded "when it is big enough." If the embedding verb is *think*, which takes a CP, then the derivation goes like this:

	Workspace:	Clock:
(4)	{ Bill, think, he, lose } →	vP
	{[Bill think]$_{vP}$, [he lose]$_{vP}$} →	TP
	{[[Bill thinks]$_{vP}$]$_{TP}$, [[he lost]$_{vP}$]$_{TP}$} →	CP
	{[[[Bill thinks]$_{vP}$]$_{TP}$]$_{CP}$, [that [[he lost]$_{vP}$]$_{TP}$]$_{CP}$}→	embedding
	{[[[Bill thinks [that[[he lost]$_{vP}$]$_{TP}$]$_{CP}$]$_{vP}$]$_{TP}$]$_{CP}$ }	

The embedding takes place at the very end, when the embedded clause is a CP. On the other hand, if the embedding predicate is *see*, which takes a vP-sized clause, then the derivation goes as follows:

	Workspace:	Clock:
(5)	{ Bill, see, him, lose } →	vP
	{[Bill see]$_{vP}$, [him lose]$_{vP}$} →	embedding
	{[Bill see [him lose]$_{vP}$]$_{vP}$} →	TP
	{[[Bill saw [him lose]$_{vP}$]$_{vP}$]$_{TP}$} →	CP
	{[[[Bill saw [him lose]$_{vP}$]$_{vP}$]$_{TP}$]$_{CP}$} →	

As these derivations show, the embedding of the complement of *see* is "earlier" than the embedding of the complement of *think*. Williams 2003 called this timing of embedding in terms of F-structure "Level Embedding"; I will refer to MGET regimented by Level Embedding as simply "Level Embedding."

Level Embedding answers all questions about the interaction of embedding of clauses with each other and with other operations, and so meets the *Aspects* objections to generalized embedding transformations. First, the ordering of different embeddings with one another is fixed: two CP clauses will be embedded simultaneously, but a vP-sized clause will be embedded before a CP-sized clause. Second, the interaction of embedding with other operations is fixed as well, because now F-structure times all operations. The embedding of a clause of size F_iP will follow all operations that target F_{i-1} and smaller, and it will precede all operations that target F_{i+1} and larger. So, all interactions are fixed by the theory, and Chomsky's objection no longer holds. It is a fact that singular operations do apply to the embedding clause before the embedding takes place, and Chomsky claimed that "there were no really convincing cases" of that; but Level Embedding does nevertheless meet the spirit of Chomsky's objection, because all ordering interactions are forced. And, as we will see below, the ordering of singularies to the embedding clause before the embedding has great explanatory potential after all.

So, Level Embedding is immune to Chomsky's objections to generalized embedding operations, but is there any reason to prefer it to the scheme of derivation that we have called BT? Williams 2003 demonstrates that some general patterns of explanation are directly derivable from Level Embedding but not available in BT. These are first, a family of locality constraints, and second, a generalized "improper movement" constraint.[6] In both cases, the explanations arise from the fact that under Level embedding, F-structure times everything—Merge, movement, clausal size, and clausal embedding.

To illustrate the kind of locality condition that follows from Level Embedding, we will consider subject raising of the sort that occurs with verbs like *seem*. It is unexplained in BT that subject raising is impossible from indirect questions, in that there seems to be no verb which takes an indirect question complement and allows subject raising out of it. In the following, *queem* is a made-up verb of the non-existing type:

(6) *John$_i$ queems [whether t$_i$ to go]

Level Embedding tells us that such predicates are impossible. It comes down to the ordering of two F-structure points: the point at which embedding takes place, and the point at which raising takes place. The point at which embedding takes place for questions must be the point that we have identified as CP, as questions always have a C; so, before that point, the structure of (6) is the workspace below, with the clauses still separate:

(7) { [(it) queems]$_{TP}$, [John to go]$_{TP}$ }

In order for the embedding to take place, CPs must be built up, since *queem* requires a CP:

(8) (7) → (CP step)
{ [[it queems]$_{TP}$]$_{CP}$, [whether[John to go]$_{TP}$]$_{CP}$ } → (embedding step)
{ [[it queems [whether [John to go]$_{TP}$]$_{CP}$]$_{TP}$]$_{CP}$ }

The other relevant F-structure point is the point at which raising to subject takes place. Since raising targets SpecT, that point is the F-structure element we have identified as TP. This means that at the time at which raising must take place the sentence has the structure in (7); but raising cannot take place in (7), because the embedding has not occurred yet. Once the embedding has taken place, as in (8, last line), it is too late to do raising, as the derivation has passed on to point

[6] The model also directly provides a generalized "reconstruction relation" not directly relevant here; see Williams 2003 for explanation.

CP. The net result is that raising must always be from a TP, never a CP, and that is why indirect questions are always islands for raising.

We can easily generalize this kind of locality: if a movement targets F_i, then clauses larger than F_i will be islands for that movement. This follows immediately from Level Embedding; BT makes no such prediction.

In a similar way, Level Embedding rules out "improper movements" of the kind represented in (9):

(9) *[John$_i$ seems [t_i [t_i to go]$_{TP}$]$_{CP}$]$_{TP}$

That is, a movement cannot relate the SpecC of a lower clause to the SpecT of an upper clause, even though the upper SpecT c-commands the lower SpecC.

This again follows immediately from Level Embedding: in order for the illicit movement to take place, the embedding must have already been done; however, that would mean that we have reached CP level operations, and it is then too late to perform an operation that targets specT. Again, we can easily generalize the conclusion: no movement from F_i to F_j is possible unless $F_i < F_j$, even if F_j is in a higher clause than F_i (and hence c-commands F_i). And again, BT makes no such prediction.

References

Chomsky, N. 1965. *Aspects of the Theory of Syntax*, MIT Press.
Williams, E. 1974. Rule Ordering in Syntax. PhD Dissertation, MIT.
Williams, E. 2003. *Representation Theory*. Cambridge, MA: MIT Press.