DOI: http://dx.doi.org/10.31513/linguistica.2017.v13n2a14026

SYNTACTIC STRUCTURES: FORMAL CONSIDERATIONS 60 YEARS LATER

Howard Lasnik[1]

Chomsky (1955), *The Logical Structure of Linguistic Theory* (henceforth LSLT), laid out in great detail the formal foundations for a rigorous new way of looking at language scientifically, transformational generative grammar. This awesome accomplishment was announced to the world in Chomsky (1957), *Syntactic Structures* (henceforth SS), a publication that revolutionized the field, or really, created a new field. Needless to say, syntactic theory has undergone vast changes since then, but certain fundamental ideas, and even a few technical details, persist. In this article, I will briefly discuss some instances of each sort.

The core formal syntactic theory of SS is that of LSLT, since the former was based on

Chomsky's teaching notes for the latter. Many of the differences that do arise do so just because SS is simplified, sometimes substantially, to fit the material into a small book that could be published (or, indeed, into a one semester course at MIT). The initial formal discussion in SS, centered on finite state Markov processes, is actually totally independent of LSLT. There is no discussion of such devices in LSLT. They are presented in SS as the most limited computational machine capable of capturing one of the most basic properties of human languages - discrete infinity (and because they represented technical machinery that MIT students would be familiar with). Markov processes with 'loops' can generate infinite languages. In SS, Chomsky shows the inadequacy of Markov processes, even with loops added, for infinite languages with certain kinds of recursion, in particular those with dependencies nested within dependencies of essentially the same type. When dependencies are nested within dependencies, we move beyond the bounds of finite state description. Chomsky gives some formal languages on the alphabet {a,b} as representative examples:

(1) ab, aabb, aaabbb, etc.

(2) aa, bb, abba, baab, aaaa, bbbb, aabbaa, abbbba, and, in general, all and only sentences

1 University of Maryland. E-mail: lasnik@umd.edu

consisting of a string *X* of *a*s and *b*s followed by the 'mirror image' of *X*.

Neither of these languages is finite state. Chomsky then gives some templates for fragments of English, that he suggests illustrate this kind of situation:

(3) If S_1, then S_2.
(4) Either S_3, or S_4.
(5) The man who said that S_5, is arriving today.

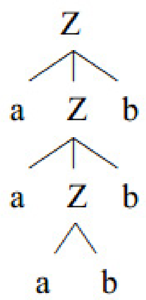
Here the dependencies are between 'If' and 'then', 'Either' and 'or', and 'man' and 'is' (cf. "men ... are"). Crucially, as Chomsky notes, each of S_1 to S_5 can contain another dependency of these types, and so on. This renders these constructions non-finite state derivable.

Chomsky at this point in SS abandons Markov description and turns to more powerful description in terms of systems of phrase structure rewrite rules (called by Chomsky Σ, F grammars), which purportedly do not suffer from the observed descriptive inadequacies. Two ironies arise in the SS presentation though. First, while (1) and (2) are, indeed, straightforwardly characterizable with Σ, F grammars, it is quite unlikely that (3)-(5) are. The way classic contextfree nested dependencies like those in (1) and (2) are generated is by having each *a* or *b* in the 'first half' of the sentence introduced along with the corresponding one in the 'second half' by exactly the same occurrence of the same rewrite rule. For instance, language (1) is generated by the following grammar, with the abstract non-terminal symbol Z.

(6) Σ: Z
F: Z → ab
F: Z → aZb

In any given sentence of the language, the first *a* is introduced simultaneously with the last *b*, then the second *a* with the penultimate *b*, and so on. This yields phrase structures such as (7).

(7)



However, it is difficult to imagine linguistically plausible structures for (3)-(5) that have the mutually dependent items as siblings, that is, simultaneously introduced by the same operation of the same rewrite rule. To the extent that this is true, these phenomena are not just beyond the bounds of finite state description, they are also beyond the bounds of Σ, F description. This specific inadequacy lies in the realm of what Chomsky (1965) called 'strong generative capacity' (strings and their associated structures).

The second irony involves a somewhat arbitrary limitation that LSLT imposes on Σ, F modules of human language grammars and that is carried over in SS. The theory of human language grammar that Chomsky assumes in SS, following the one articulated in LSLT, restricts the power of the Σ, F module in precisely such a way that it cannot even in principle handle the phenomena discussed just above. In particular, the theory explicitly and completely disallows recursion in this module (pp. 517-519). In this model, the infinitude of human languages is the responsibility of generalized transformations - operations melding separate phrase markers together into one phrase marker. Though LSLT had indicated that the restriction constituted a simplification, Chomsky didn't actually offer any arguments to that effect. One might actually argue that removing this restriction is a simplification. After all, it seems to be a stipulation.

Further, while trivial, it is not always simple to determine whether there is recursion in the base.

Certainly, the determination is simple if there is a rule like (8):

(8) A → BA

But recursion might involve a pair of rules rather than any one rule:

(9) A → BC
C→ DA

Or a trio:

(10) A → BC
C→ DE
E → FA

In fact, there is no limit on how large the minimal group of rules might be that yields recursion.

Chomsky (1965), *Aspects of the Theory of Syntax*, claimed, contrary to his position in LSLT, that the theory of transformational grammar is simplified by *allowing* recursion in the Σ, F component, the simplification being that the notion 'generalized transformation' is eliminated entirely, at no apparent cost. Thus, in place of three kinds of syntactic operations - Σ, F rules, singulary transformations (those operating on a single 'tree'), and generalized transformations we have just the first two. Further, the construct 'Transformation-marker' (a history of the transformational derivation) is eliminated, as its major work was to show exactly how the separate trees combine into one, but now that is transparently represented in the initial phrase marker, the 'deep structure'. Of course in Minimalism, generalized transformations (instances of 'external merge') are back with a vengeance. They are now responsible for almost **all** structure building, not just the combination of sentential phrase markers. The reply to the Chomsky (1965) simplicity argument is that we still just have two kinds of syntactic operations: generalized transformations (external merge) and singulary transformations ('internal merge'). Σ, F rules are gone. In fact, the situation might be better still. Chomsky (1995) suggested in his footnote 13 that the two kinds of operations are both instances of the same basic operation, Merge. This has become a standard view, so we are down from three syntactic operations to one.

Another argument offered by Chomsky (1965) against the generalized transformations approach of LSLT/SS is also still of current import. Chomsky claimed that while there are many cases of singulary transformations that must apply to a constituent sentence before it is embedded, or that must apply to a 'matrix' sentence after another sentence is embedded in it, "there are no really convincing cases of singulary transformations that must apply to a matrix sentence before a sentence transform is embedded in it . . ." Given the Aspects modification, with recursion in the base, the list of transformations is claimed to apply 'cyclically,' first operating on the most deeply embedded clause, then the next most deeply embedded, and so on, working 'up the tree' until they apply on the highest clause, the entire generalized phrase marker. Thus, singulary transformations apply to constituent sentences 'before' they are embedded, and to matrix sentences 'after' embedding has taken place. "The ordering possibilities that are permitted by the theory of Transformation-markers but apparently never put to use are now excluded in principle" (1965, 135). So how can it be that Chomsky (1993) argued for generalized transformations as the sole structure creating operation, responsible even for the structure of single clause sentences? What of the powerful Chomsky (1965) against such a model, that it allowed derivations that never actually occur in human languages? The model with recursion in the base excluded those unwanted derivations. However, on closer inspection, it was not actually elimination of generalized transformations that had this limiting effect. Rather, it was the stipulation that transformations operate strictly cyclically, starting on the most deeply embedded clause and proceeding monotonically up the tree. Chomsky (1993) observed that a condition with the same effect can be imposed on the operation of generalized transformations and their interaction with singulary transformations. This condition, often called the 'Extension Condition', simply requires that a transformational operation 'extends' the tree upwards. This guarantees the same sort of monotonic derivations as those authorized by Chomsky (1965). More on this condition below.

The SS presentation of formal limitations of finite state description included, in addition to (1) and (2) above (classic context free languages), a language that is not Σ, F generable, i.e., is not context free:

(11) aa, bb, abab, baba, bbbb, aabaab, abbabb, ..., and in general, all sentences consisting of a string *X* of *a*'s and *b*'s followed by the identical string *X*, and only these.

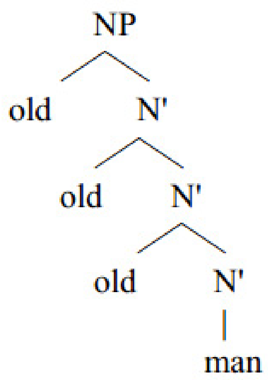
Here, in place of the unbounded nested dependencies we saw above, we have unbounded crossserial dependencies, exactly what context free rewriting systems cannot handle. The fact that some cross-serial dependencies do arise in English (in particular, in the system of verbal inflectional morphology) provides part of the motivation in SS for adding transformations to the theory of syntax, where a transformation relates one phrase structure representation to another. Chomsky acknowledges that since this portion of English syntax is finite, a Σ, F grammar or even a Markov process is, of course trivially, sufficient if all we are concerned with is the brute force enumeration of the strings, but the resulting description would be extremely complex and unenlightening, missing fundamental generalizations.

It is interesting to observe a striking difference in the SS treatments of Markov process limitations and Σ, F limitations. In the latter instance, the response was to add onto the insufficient device another device, transformations, so the resulting theory has both devices. Not so for the former, in which case the insufficient device is simply banished, possibly not the correct move, as discussed in Lasnik (2011) and Lasnik and Uriagereka (2012). As discussed in those works, one of the major benefits of Σ, F description is that, unlike finite state description, it automatically and unavoidably provides sentences with structure. This is overwhelmingly positive since, alongside infinitude, constituent structure is the most fundamental and universal property of human languages. But there are rare exceptions, as discussed by Chomsky (1961,p.15) and Chomsky and Miller (1963, p.298). One of the most striking ones is what Chomsky called "true coordination" as in (12).

(12) The man comes / The old man comes / the old old man comes / ...

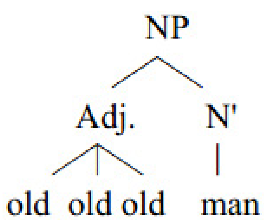
Chomsky states, for this and for certain other cases, "Immediate constituent analysis has been sharply and, I think, correctly criticized as in general imposing too much structure on sentences." That is, there is no evident syntactic, semantic, or phonological motivation for a structure in which, say, each old modifies the remaining sequence of olds plus man, as in (13), or some such (with irrelevant details omitted).

(13)



Preferable might be something like:

(14)



Chomsky says

> "The only correct P-marker would assign no internal structure at all within the sequence of coordinated items. But a constituent structure

grammar can accommodate this possibility only with an infinite number of rules; that is, it must necessarily impose further structure, in quite an arbitrary way." [p. 15]

Chomsky and Miller (1963, p.298) present a very similar argument: "... a constituent-structure grammar necessarily imposes too rich an analysis on sentences because of features inherent in the way *P*-markers are defined for such sentences."

The conclusion of Chomsky and of Chomsky and Miller: We need to go beyond the power of Σ, F description to adequately describe natural languages. In particular, the model is augmented by a transformational component.

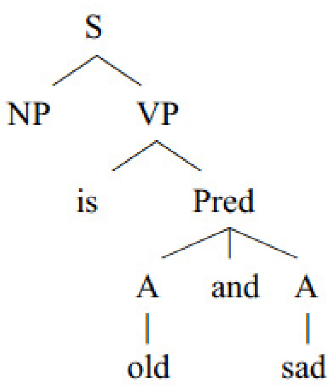
LSLT Chomsky (1955) had, of course, already shown in great detail how transformations can provide natural accounts of phenomena that can only be described in cumbersome and unrevealing ways (if at all) by Σ, F grammars. But Chomsky had little to say there about the "too much structure" problem we are now considering. Chomsky (1961) and Chomsky and Miller (1963) don't have a lot to say either, beyond the implication that transformations will solve the problem. That is, we need to move up the power hierarchy. In fact, as already mentioned, Chomsky (1955) had already claimed that there is no recursion in the Σ, F component, the transformational component (in particular generalized transformations (GTs)) being responsible in toto for infinitude.

Chomsky discussed several aspects of the coordination process, though without actually giving a precise formulation of the relevant transformation(s). It is interesting to note that all the examples discussed in Chomsky (1955) involve coordination of two items, as in (15).

(15) John was old and sad

For such cases, it is straightforward to formulate an appropriate generalized transformation, even if, as claimed by Chomsky (1961, p.134), GTs are strictly binary (an idea that is important in Chomsky's recent work, as in this quote, "arguably restriction of computational resources limits n for Merge to two", from Chomsky (2008)). The structure of (15) might then be as in (16).

(16)



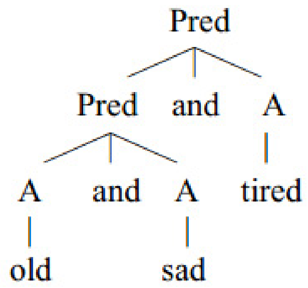
Chomsky and Miller also seem to assume binarity, at least in one place in their discussion: "The basic recursive devices in the grammar are the generalized transformations that produce a string from a pair [emphasis mine] of underlying strings." [p. 304]

It is not entirely clear what is supposed to happen when we have multiple items coordinated, as in the phenomena principally under discussion here, or in, e.g.:

(17) old and sad and tired

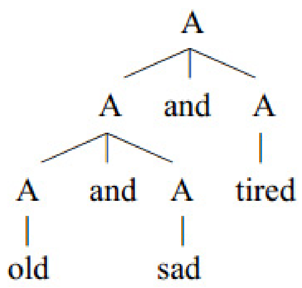
One possibility is that we would preserve the structure of "old and sad" in (16), and create a higher structure incorporating "and tired".

(18)



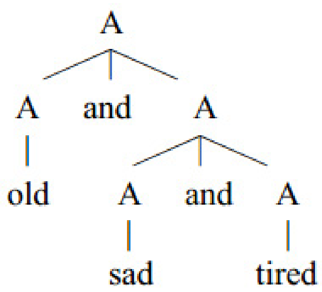
Or, somewhat revising (16):

(19)



Another possibility is a right branching analogue:

(20)



But any of these would run afoul of Chomsky's argument: In general, we do not want that extra structure.

Yet another possibility, one that would yield the desired 'flatness', arises if we relax the binarity requirement. Chomsky and Miller seemingly countenance this possibility in at least one place in their discussion: "We now add to the grammar a set of operations called *grammatical transformations*, each of which maps an *n*-tuple [emphasis mine] of *P*-markers ($n \geq 1$) into a new *P*marker." [p. 299]

Then a GT could be formulated to coordinate three items (alongside the GT coordinating two items). But, as already noted, there is no limit on the number of items that can be coordinated, which was Chomsky's original point. So this solution merely replaces one untenable situation with another: In place of an infinite number of phrase structure rules, one for each number of coordinated items, we have an infinite number of generalized transformations.

Thus, moving up the power hierarchy ultimately does not help in this instance. In a manner of speaking, what we really want to do is move down the hierarchy. Finite state Markov processes can give flat objects, since their productions are essentially concatenations. But that is not quite the answer either. While it would work fine for coordination of terminal symbols, phrases can also be coordinated, and, again, with no upper bound. Alongside (21), we find (22).

(21) John and Bill and Fred and ...

(22) The old man and the young man and the boy and ...

We need a sort of higher order flatness.

Chomsky and Miller [p. 298] consider, but reject, an extension of constituent structure grammar to yield such flatness. Their extension is, as far as I can tell, equivalent to the so-called Kleene-* device of Kleene (1956). The specific instance they give is:

(23) Predicate $\rightarrow$ Adj^n and Adj ($n \geq 1$)

Chomsky and Miller indicate that there are "many difficulties involved in formulating this notion so that descriptive adequacy may be maintained ..." But they do not elaborate on this point. I will leave this issue for further discussion elsewhere.

A transformational derivation begins with an initial phrase structure representation, or a set thereof. What is an initial phrase structure representation? Though this aspect of the formal machinery of LSLT is not stated explicitly in SS, the model is set theoretic (rather than, say, graph theoretic). The initial phrase structure representation of a sentence U (its 'phrase marker') given a particular Σ, F grammar G is constructed as follows. The Σ, F grammar comprises a designated initial symbol, or set thereof, (Σ), and a set of rewrite rules (F), which consist of one symbol on the left, followed by an arrow, followed by at least one symbol. Symbols that appear on the left side of arrows are non-terminal symbols. Those that appear only on the right are terminal symbols. It is important to note that all of these symbols are atomic. Analysis into distinctive features, already standard in phonology at the time, and so important in modern syntactic theory, was not present in syntactic theory until the 1970s. Even phrasal symbols like NP are atomic. Contrary to appearances, there is no *N* in *NP*, or *V* in *VP*, etc. Many structural generalizations were missed as a consequence, a defect not really rectified until Chomsky (1970). On the other hand, the basic idea that the primitives of syntactic theory are necessarily **syntactic** (sometimes called 'autonomy of syntax' was already present in SS and persists to this day.

A phrase structure *derivation* consists of a series of lines such that the first line is one of the designated initial symbols, and the procedure for moving from one line to the next is to replace exactly one symbol by the sequence of symbols it can be rewritten as. For all but the most trivial grammars, there will be multiple *equivalent* derivations for any particular string, where two derivations are equivalent if and only if they involve the same rewrite rules the same number of times, but not necessarily in the same order. The phrase marker of the produced terminal string is the set of all strings occurring in any of the equivalent derivations. For illustrative purposes, I give a toy example:

(24) Σ: S
F: S → NP VP   NP → Mary   VP → V   V → laughs

This grammar generates one sentence:

(25) Mary laughs

The equivalent derivations of (25) are in (26).

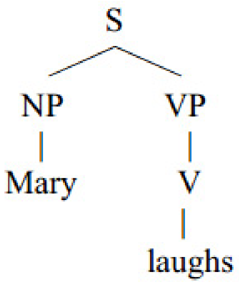
(26)			
	S	S	S
	NP VP	NP VP	NP VP
	Mary VP	NP V	NP V
	Mary V	NP laughs	Mary V
	Mary laughs	Mary laughs	Mary laughs

The phrase marker (PM) is (27).

(27) {S, NP VP, Mary VP, Mary V, NP V, NP laughs, Mary laughs}

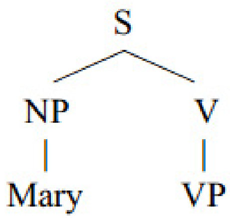
This suffices to capture what LSLT and SS took to be the essence of phrase structure, the 'is a' relations between portions of the terminal string and single non-terminal symbols. *Mary* is an NP, *laughs* is a V and a VP, *Mary laughs* is an S. A familiar graphic representation of this PM is the phrase structure tree in (28).

(28)



It is worth noting that a set theoretic representation of the sort in (27) has strictly less information than a graph like (28). For instance, the set is neutral between several graphs, including (28) and (29).

(29)



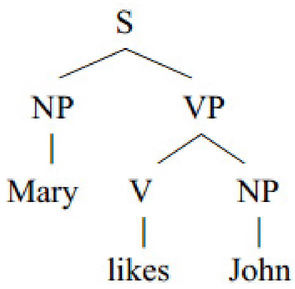
I will return to this difference between the two kinds of representations. But first, another difference. Consider a slightly augmented grammar and a slightly more complicated sentence:

(30) Σ: S
F: S → NP VP   NP → Mary   NP → John   VP → V   VP → V NP   V → laughs   V → likes

(31) Mary likes John

In familiar tree form, we have

(32)



Notice that a constituent, e.g. VP here, is a sub-structure of the whole structure, a sub-tree of the tree. Consider now the LSLT/SS type set theoretic PM for (31) given grammar (30):

(33) {S, NP VP, Mary VP, NP V NP, Mary V NP, NP V John, Mary V John, NP likes NP, Mary likes NP, NP likes John, Mary likes John}

A set representing the phrase structure of the VP might be (34).

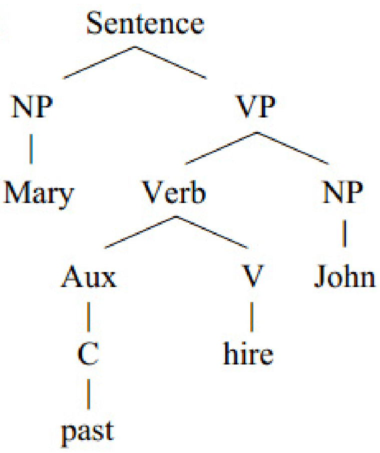
(34) {VP, V NP, V John, likes NP, likes John}

But there is no subset of (33) even remotely resembling (34). This case is completely representative. Thus, as discussed by Lasnik and Stone (2016) and by Stone (2017), the notion 'sub-structure' is surprisingly difficult to capture in a set-theoretic model like that in LSLT/SS, or the somewhat revised one in Lasnik and Kupin (1977).

The presentation of transformations in SS is rather informal. Especially with respect to 'structural change' (what the transformation does), it is just shorthand, not actually specifying, as the fuller presentation in LSLT does, the particular operation being performed, but rather just the revised sequence of terms. For expository purposes, I present here a slightly simplified version of a tiny portion of the SS analysis of English verbal inflectional morphology (for my money, the single best syntactic analysis of anything ever), beginning with a fragment of the Σ, F module, then proceeding to the two of the transformations, where the *terms* are set off by hyphens. Before proceeding I want to point out some of the major differences between syntactic theory of the mid-1950s and that 60 years later. First, there was no lexicon in the sense of Chomsky (1965), pretty much still the modern sense. Lexical items were introduced in exactly the same way that non-terminal symbols were - via the Σ, F component. Second, the structure building via the Σ, F component was basically 'top down', beginning with rewriting of the designated initial symbol. This contrasts with the 'bottom up' bare phrase structure approach, which begins with items pulled from the lexicon. Third, a grammar could consist of an unlimited number of transformations, as there is no bound on the number of terms in a structural analysis (the part of a transformation that specifies what phrase markers or derived phrase markers are eligible to undergo the transformation). This sharply contrasts with principles and parameters theorizing, culminating in Minimalism. Other differences will emerge in the following discussion.

(35) Σ: Sentence
F:
Sentence → NP VP
NP → John [Simplifying, to keep the set theoretic phrase marker (PM) from getting too unwieldy]
NP → Mary
VP → Verb NP
Verb → Aux V
Aux → C (Modal) (have en) (be ing)
C → past [Simplifying again, for the same reason]
V → hire

(36)



(37) {S, NP VP, NP Verb NP, NP Aux V NP , NP C V NP, NP past V NP, John VP, etc., etc., etc.}

(38) T_{not} - optional
Structural analysis:

$$\left\{ \begin{array}{l} \text{NP - C - V...} \\ \text{NP - C+M - ...} \\ \text{NP - C+have - ...} \\ \text{NP - C+be - ...} \end{array} \right\}$$

Structural change: $X_1 - X_2 - X_3 \rightarrow X_1 - X_2 + n't - X_3$ [This is intended to be right adjunction of *n't* to the 2nd term of the SA.]

(39) T_q - optional [Interrogation "Subject Aux Inversion"]
Structural analysis: same as T_{not}
Structural change: $X_1 - X_2 - X_3 \rightarrow X_2 - X_1 - X_3$ [This is intended to be permutation of the 1st and 2nd terms of the SA.]

The SA of (38) and (39) (and a third related T as well) is interestingly different from that of the large majority of the transformations in SS and LSLT. Overwhelmingly, the transformations display a property later called 'structure dependence', a property still fundamental in Chomskian syntactic analysis. In particular, it is *constituents*, units of structure, that are manipulated. This is why SS claims that "... the behavior of a sentence under transformation provides valuable, even compelling evidence as to its constituent structure." [p.81] A fundamental way of enforcing structure dependence is to require that the terms in the SA of a transformation are single symbols. And generally they are, but this set of transformations do not conform. For the second, third, and fourth disjuncts, the second term is a <u>sequence</u> of two symbols. Beginning in the mid 1960's, and continuing to the present, a variety of proposals appeared that had the effect of combining the first auxiliary verb with the tense node into a single constituent, often by the first auxiliary verb adjoining to Tense (C of SS). That single constituent would then be the target of transformations in a structure dependent way.

Since all the examples we have been considering involve the application of multiple transformations, it is important to indicate how this can be possible. This question arises because transformations are defined to apply to phrase markers, and phrase markers emerge from Σ, F derivations. Thus, once a transformation has applied, we no longer have a phrase marker. If we are ever to be able to construct a non-trivial transformational derivation, the result of applying a transformation must then be of just the same formal character as a phrase marker. There are principles of derived constituent structure in LSLT (presupposed in SS) that guarantee this.

The derivations in LSLT/SS generally have a strongly Markovian character, in that the applicability of a given transformation depends only on the current (derived) phrase marker, just as in current syntactic theories. The set of transformations constitute a strict linear ordering, successively altering the phrase marker. But there is occasional 'globality'. For instance, T_W, which is responsible for WH-interrogatives, is flagged as "conditional on T_q". This means that even if the current PM satisfies the SA of the T, the T can only apply if earlier in the derivation T_q had applied. This is needed to rule out (40).

(40) *Who Mary will hire [cf. Who will Mary hire]

The T-marker can provide this kind of information, but it is very seldom called upon.

Interestingly, six decades later, I'm not sure there is a really satisfying analysis of (40).

The transformational component in SS seems somewhat ancient in consisting of many specific strictly ordered transformations, each stipulated as optional or obligatory, rather than the modern one or two very general ones (optional in GB, obligatory, in essence, under Minimalism). The rule ordering is, from a modern perspective, especially unfamiliar. The rule ordering is stipulated; each transformation has a fixed position in the linear ordering, and this doesn't follow from any intrinsic properties of the particular rules. The next language over might have the very same transformations but in a different fixed order. Needless to say, this represented a massive learnability problem, one ultimately eliminated in work of the 1970s, with stipulated orderings banished in favor of general ordering principles, including the principle of cyclic application mentioned above, introduced into syntactic theory in Chomsky (1965), though, interestingly, it already had appeared in phonological theory a decade earlier in Chomsky et al. (1956). The Minimalist version of the cyclic principle is the Extension Condition, the requirement that Merge, **the** transformation, attaches material only at the 'top of the tree'. It is somewhat surprising to realize that in the LSLT/SS model, movement of an item already in the structure to a new position at the top of that structure is, literally impossible. Transformations in that framework consist of a Structural Analysis (SA) and a Structural Change (SC). The former determines whether the T is applicable to a particular PM, while the latter indicates the operation to be performed. An SA is a sequence of `terms', each term a (string) variable, a constant (i.e., a syntactic symbol), or a linear combination of any of the preceding. Applicability is determined by comparing the SA with the members of the phrase marker set to establish satisfaction. Any string satisfies a variable, while a constant is satisfied only by that very symbol. Notice that every member of any PM has symbols in a linear order; every pair of symbols in a member are in the precedence relation. Thus, the symbols in any SA are likewise necessarily in a linear order. Thus, as discussed by Lasnik and Stone (2016), a symbol can adjoin to one that follows it or to one that precedes it. An operation that would adjoin a symbol to a dominating symbol is literally unstatable. But any singulary movement T satisfying the Extension Condition would have to do exactly this.

On the other hand, the modular character of the transformations has a bit of a more modern feel. For example, the interrogative and negation transformations given above are in part, but only in part, responsible for sentences like "Did Mary hire John" and "Mary didn't hire John". This is so because those Ts set the stage for the insertion of supportive *do*, but don't in themselves effect it. There is a separate transformation for that. This means that all of the mentioned Ts can be kept relatively simple. Further, a huge generalization is captured: Whenever Affix Hopping is blocked, supportive *do* appears. Additional cases are the elliptical "Susan didn't hire John but Mary did ~~hire John~~" and the WH-interrogative "Who did Mary hire". Another instance of this kind of simplifying modularity is negative questions like "Didn't Mary hire John". There is no negative question transformation, because we get such sentences for free by the interaction of the independently necessary interrogative, negation, and *do* transformations. All of these interestingly contrast with the strongly non-modular treatment of passive sentences. There are three clusters of properties distinguishing simple English passives from actives: passive morphology on the main verb, thematic object appearing in subject position, and displacement of subject into a PP headed by *by*. The Passive T in SS performs all these operations. Chomsky (1965) took steps towards modularizing passives, and Chomsky (1970) went much further. Chomsky (1981) completed the process.

The theoretical edifice built on the LSLT/SS foundation has undergone extensive revision over the years. Some of that edifice is hardly recognizable anymore. But much of it remains. All of the following are as important to transformational grammar as they were at their introduction sixty years ago: recursion; structure; abstract underlying structure; singulary and generalized transformations; derivations; autonomy of syntaax. Any syntactician has much to gain from carefully considering that foundation.

REFERENCES

Chomsky, Noam. 1955. The logical structure of linguistic theory. Ms. Harvard University, Cambridge, Mass. and MIT, Cambridge, Mass. [Revised 1956 version published in part by Plenum, New York, 1975; University of Chicago Press, Chicago, 1985].

Chomsky, Noam. 1957. *Syntactic structures*. The Hague: Mouton.

Chomsky, Noam. 1961. On the notion "rule of grammar". In *Structure of language and its mathematical aspects*, ed. Roman Jakobson, 6-24. Providence, Rhode Island: American Mathematical Society. [Reprinted in The structure of language: Readings in the philosophy of language, eds. Jerry A. Fodor and Jerrold J. Katz, 119-136. Englewood Cliffs, N. J.: Prentice-Hall, 1964].

Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge, Mass.: MIT press.

Chomsky, Noam. 1970. Remarks on nominalization. In *Readings in English transformational grammar*, ed. Roderick A. Jacobs and Peter S. Rosenbaum, 184-221. Waltham, Mass.: Ginn and Co. [Reprinted in Noam Chomsky, *Studies on semantics in generative grammar*, 11-61. The Hague: Mouton, 1972].

Chomsky, Noam. 1981. *Lectures on government and binding*. Dordrecht: Foris.

Chomsky, Noam. 1993. A minimalist program for linguistic theory. In *The view from Building 20: Essays in linguistics in honor of Sylvain Bromberger*, ed. Kenneth Hale and Samuel J. Keyser, 1-52. Cambridge, Mass.: MIT Press. [Reprinted in Noam Chomsky, *The minimalist program*, 167-217. Cambridge, Mass.: MIT Press, 1995].

Chomsky, Noam. 1995. Bare phrase structure. In *Government and binding theory and the minimalist program*, ed. Gert Webelhuth, 383-439. Oxford: Basil Blackwell. [Also in *Evolution and revolution in linguistic theory: Essays in honor of Carlos Otero*, eds. Hector Campos and Paula Kempchinsky. Washington D. C. : Georgetown University Press].

Chomsky, Noam. 2008. On phases. In *Foundational issues in linguistic theory*, ed. Robert Freidin, Carlos Otero, and Maria-Luisa Zubizarreta. Cambridge, Mass.: MIT Press.

Chomsky, Noam, Morris Halle and Fred Lukoff. 1956. On accent and juncture in English. In *For Roman Jakobson*, ed. M. Halle, H. Lunt, and H. MacLean, 65-80. The Hague: Mouton & Co.

Chomsky, Noam and George Miller. 1963. Introduction to the formal analysis of natural languages. In *Handbook of Mathematical Psychology 2*, ed. Duncan R. Luce, Robert R.

Bush, and Eugene Galanter, 269-321. New York: Wiley and Sons.

Kleene, Stephen. 1956. Representation of events in nerve nets and finite automata. In *Automata Studies*, ed. Claude Shannon and John McCarthy, 3-41. Princeton: Princeton University Press.

Lasnik, Howard. 2011. What kind of computing device is the human language faculty? In *The Biolinguistic Enterprise: New Perspectives on the Evolution and Nature of the Human Language Faculty*, ed. Anna Maria Di Sciullo and Edwin Williams, 354-365. Oxford: Oxford University Press.

Lasnik, Howard and Joseph J. Kupin. 1977. A restrictive theory of transformational grammar. *Theoretical Linguistics* 4: 173-196. [Reprinted in *Essays on restrictiveness and learnablity*, Howard Lasnik, 17-41. Dordrecht: Kluwer, 1990]

Lasnik, Howard and Zach Stone. 2016. Rethinking phrase structure. Ms. University of Maryland, College Park, MD

Lasnik, Howard and Juan Uriagereka. 2012. Structure. In *Handbook of Philosophy of Science Volume 14: Philosophy of Linguistics.*, ed. Ruth Kempson, Tim Fernando, and Nicholas Asher, 33-61. Amsterdam: Elsevier.

Stone, Zach. 2017. The spatial structure of minimalist syntax. Ms. University of Maryland, College Park, MD