On the mathematical foundations of *Syntactic Structures*

Geoffrey K. Pullum
School of Philosophy, Psychology and Language Sciences
University of Edinburgh

5 April 2011

Abstract Chomsky's highly influential *Syntactic Structures* (*SS*) has been much praised its originality, explicitness, and relevance for subsequent cognitive science. Such claims are greatly overstated. *SS* contains no proof that English is beyond the power of finite state description (it is not clear that Chomsky ever gave a sound mathematical argument for that claim). The approach advocated by *SS* springs directly out of the work of the mathematical logician Emil Post on formalizing proof, but few linguists are aware of this, because Post's papers are not cited. Chomsky's extensions to Post's systems are not clearly defined, and the arguments for their necessity are weak. Linguists have also overlooked Post's proofs of the first two theorems about effects of rule format restrictions on generative capacity, published more than ten years before *SS* was published.

Keywords Generative grammar · transformations · Emil Post · formalization · proof theory · mathematical logic

1 Introduction

In 1957, when Martin Joos first published his classic anthology of American structuralism, *Readings in Linguistics I* (1957), it was offered as a contribution to a flourishing research program. The same could be said about J. R. Firth's collected *Papers in Linguistics 1934-1951* (Firth 1957) and the Philological Society's definitive anthology of London-school linguistics, *Studies in Linguistic Analysis* (Philological Society 1957), or about B. F. Skinner's long-awaited book about verbal behavior based on his William James Lectures from ten years earlier (Skinner 1957). In retrospect, however, these works look like valedictions: concluding summaries of paradigms that had reached their use-by dates.

In December 1957, *Language* published an extraordinarily laudatory review (Lees 1957) of a short monograph that had been published on February 14 that year in a new series from a small publisher in Holland. The book was *Syntactic Structures* (Chomsky 1957, henceforth *SS*). The author was at the time an unknown 28-year-old who taught language classes at MIT. The reviewer, Robert B. Lees, hailed *SS* as a revolutionary scientific breakthrough, and from 1958 on, linguists began to pay it a great deal of attention.

Lees's claims about revolution are controversial. Newmeyer (1986) argues that *SS* did indeed spark a scientific revolution, and others disagree. I take no position here on that question (a sociological one, as Newmeyer construes it). I will argue, however, that many exaggerated claims have been made about *SS*, some of them straightforwardly false. For example, the claim that in *SS* Chomsky gave a "proof" that "demonstrated the inadequacy

---

This paper has been published online by the *Journal of Logic, Language and Information* (at the http://www.springerlink.com website; DOI 10.1007/s10849-011-9139-8), and will subsequently appear in the print edition. This is my own text, differeing in pagination from the version published by *JoLLI*. To contact the author: E-mail: gpullum@ling.ed.ac.uk or phone +44-131-650-3603.

of finite state grammars" (Lyons 1970: 54) is not true: Chomsky did not even attempt such a proof in *SS* (see section 2 below).

Recent claims about the content and effects of *SS* have been getting more extreme rather than less. Josie Glausiusz says that Chomsky "in the 1950s proposed that all humans are equipped with a universal linguistic grammar, a set of instinctive rules that underlie all languages" (*Discover* magazine, May 2007). But Chomsky made no proposals about innate universal grammar in the 1950s. David Lightfoot calls *SS* "the snowball which began the avalanche of the modern 'cognitive revolution' [that] originated in the seventeenth century and now construes modern linguistics as part of psychology and human biology" (Lightfoot 2002: *v*). But *SS* contains not even a nod in the direction of the study of cognition or 17th-century thought.

When the reputation of a book gives rise to this kind of exaggeration and misattribution, re-evaluation is called for. The task is a large one, and I do not aim to cover all aspects of *SS* in this paper.[1] My focus is on selected aspects of the mathematical and formal language-theoretic underpinnings of *SS*.

2 The purported proof that English is not finite-state

It is widely believed that *SS* gives a proof that the stringset of all syntactically well-formed sequences of English words is not a finite-state language (FSL). It does not. *SS* never attempts a rigorous argument; it just informally adumbrates one, and then asserts that "it is *impossible*, not just difficult, to construct a device of the [finite automaton] type ... which will produce all and only the grammatical sentences of English" (*SS*: 23). A weak generative capacity proof is required to show that English is not an FSL, and *SS* did not give one.

Of course, *SS* was deliberately trying to keep things elementary—it originated as the notes for a series of lectures for undergraduate scientists and engineers at MIT, and was supposed to offer an informal digest of a technical paper, Chomsky (1956b), and a much larger unpublished typescript, Chomsky (1956a, published much later with revisions and omissions as Chomsky 1975). But in fact the earlier works do not contain a demonstration of the non-FSL character of English either.

SS (p. 22) cites Chomsky 1956b, claiming it contains the "rigorous proof" to which *SS* alludes, but this is not so. In the form originally given in Chomsky (1956b) it depended on a cumbersomely defined relation of "(i, j)-dependency" holding between a string S of length n, two integers i and j such that $1 \leq i < j \leq n$, and a language L over a vocabulary A, and the attempted argument was not valid. The definitions are changed in the 1965 reprint version of the paper (a footnote credits E. Assmuss for pointing out a defect in the original formulation): this revised version relies on a cumbersomely defined ternary relation of 'm-dependency' between a sentence S, an integer m, and a stringset L, where $S = x_1\, a_1\, x_2\, a_2\, \ldots\, x_m\, a_m\, z\, b_1\, y_1\, b_2\, y_2\, \ldots\, b_m\, y_m$, and there is a unique mapping of the set $\{1, \ldots, m\}$ to itself (a "permutation") meeting the following condition (I quote from p. 108):

> there are $\{c_1, \ldots, c_{2m}\} \in A$ such that for each subsequence $(i_1, \ldots, i_p)$ of $(1, \ldots m)$, S_1 is not a sentence of L and S_2 is a sentence of L, where
> (10) S_1 is formed by substituting c_{i_j} for a_{i_j} in S, for each $j \leq p$;
> S_2 is formed by substituting $c_{m+\alpha(i_j)}$ for $b_{\alpha(i_j)}$ in S_1, for each $j \leq p$.

The idea is that if in the string S the symbol a_i is replaced by the symbol c_i, restoring grammaticality in L necessitates replacing $b_{\alpha(i)}$ by $c_{m+\alpha(i)}$.

From there, the crucially relevant mathematical step is to claim that an FSL can only exhibit m-dependencies up to some finite upper bound on m (Chomsky says an m-dependency

[1] For example, I will not discuss here the critique of statistical approaches to grammaticality (*SS*: 15–17), though it was so influential that the now burgeoning work on stochastic approaches to grammar virtually disappeared from the scene for thirty years. The claim that probabilistic models can never distinguish grammatical but unlikely nonsense sequences from ungrammatical sequences (*SS*: 16) was not true, but it was not properly answered until Pereira (2000) showed what a huge difference it makes when a crude statistical model that assigns zero probability to anything not yet attested is replaced by one that uses Good–Turing 'smoothing'. I also omit discussion of the philosophical and historical background; for that, see Tomalin (2006), Scholz & Pullum (2007), and Seuren (2009).

needs at least 2^m states; Svenonius 1957 says this is untrue, and m states will suffice). The empirical claim is that English has no such upper bound, and is therefore not an FSL.

But Chomsky does not complete the argument by connecting these abstractions to English data; he merely points to some sentence templates ("If S_1, then S_2"; "Either S_3, or S_4"; "The man who said that S_5, is arriving today" [comma in original]), and asserts that through them "we arrive at subparts of English with ... mirror image properties" and thus "we can prove the literal inapplicability of this model" (Chomsky 1956b, 1965 reprinting, p. 109).

Daly 1974 spends many pages attempting to work out how a sound argument for Chomsky's conclusion might be based on the data that he cites. Chomsky seems to think that pairs like ⟨*if, then*⟩ and ⟨*either*, *or*⟩ give rise to *m*-dependencies. Daly could not see how this could be true. Nor can I. The words in these pairs can occur in sentences without the other member of the pair. (The same is true of other pairs such as ⟨*neither*, *nor*⟩ and ⟨*both, and*⟩.) It is not clear that there is *any* pair of lexical items σ and τ in English such that if $\varphi\sigma\psi$ is grammatical then $\psi = \psi_1\ \tau\ \psi_2$ with $|\psi_1| > 0$.

In addition, remarks like "we can find various kinds of non-finite state models within English" (*SS*: 22–23) and the similar remark that "we arrive at subparts of English with ... mirror image properties" (Chomsky 1956b, 1965 reprinting, p. 109) suggest a failure to appreciate that FSLs can have infinite non-finite-state subsets. Only if such a subset can be extracted by some regularity-preserving operation like homomorphism or intersection with a regular set does it entail anything about the language as a whole.

Thus it is not at all clear that Chomsky ever had an argument against English being an FSL. Certainly none appears in *SS*, which contains far less on this than the 1956 paper.

The discussion in *SS* actually shows some signs of a confusion between the FSLs and the strictly local (SL) stringsets. The exposition of finite-state systems in *SS* is particularly informal: no definitions are given, grammars are not distinguished from accepting automata, finite-state Markov processes are not distinguished from their transition graphs, and it is not at all clear that Chomsky had a grasp of how rich and complex the class of FSLs is. He gives only one example of an infinite FSL: the stringset denoted by the regular expression **the old***((**man comes**) + (**men come**)). But this is not merely within the FSLs; it is in all of the infinite hierarchy of classes of stringsets that Rogers & Pullum (2010) refer to as LTO_k, for k all the way down to 2; and it is in all the LTT_k subsets of that (for k down to 2); and in all the LT_k subsets of that (for k down to 2); and in all the strictly k-local or SL_k subsets of that (for k down to 2). In other words, it is just about as low in the subregular hierarchy as any infinite stringset class that is interesting enough to study—quite extreme in its low language-theoretic complexity, and a singularly unrepresentative FSL.

The confusion is amplified by Lasnik (2000), a syntax textbook grounded in *SS*. Lasnik considers (or alludes to) six different infinite FSLs (see pp. 12–16 and exercise 1 on p. 34): (i) *the old** ((*man comes*), (*men come*)); (ii) *the man runs* (*and runs*)*; (iii) a^*b^*; (iv) $(a,b)^*$; (v) *my sister laughed* (*and laughed*)*; (vi) *Mary saw three* (*very*)* *old men*. Every one of these is an SL_2 stringset.

Confusion between the FSLs and much smaller stringset classes such as SL_2 has been problematic at various points in the subsequent literature on the psychology of language. Bever, Fodor & Garrett (1968) criticize associationist psychology by linking it to a language-theoretic property that appears to pick out the SL class, but do not expositorily distinguish that from the FSLs (see Pullum & Scholz 2007 for discussion). And experiments on syntactic pattern recognition in non-human organisms such as those conducted by Fitch & Hauser (2004) have clearly suffered from a failure to differentiate finite state from SL_2, which has given rise to statements in the literature about monkeys being able to learn arbitrary FSLs (see Pullum & Scholz 2009, Rogers & Pullum 2010).

3 The foundational work of Emil Post

With finite-state description supposedly ruled out, the plan that *SS* aims to pursue is to introduce context-free phrase structure grammars (CF-PSGs) next, and show their effectiveness in defining a range of simple clause structures, and then to show that those also have failings, and thus motivate the introduction of a new class of rules to augment CF-PSG: transformations. The combination of CF-PSG with transformations defines the "transformational model for linguistic structure” (*SS*: 6), later known as transformational-generative grammar (henceforth TGG).

TGG is generally assumed to have sprung entirely from Chomsky’s work, specifically his large unpublished manuscript (Chomsky, 1956a) and the brief undergraduate-lecture digest of it that was published as *SS*. While linguists are aware that the term “transformation” comes from the work of Chomsky’s mentor Zellig Harris, and some have noted that Harris probably took the term from Carnap (1934), it has gone almost entirely unremarked that the underlying mathematics is largely present in much earlier work, overlooked by linguists because Chomsky never cited it. The machinery that TGG employs (though not the proposals about how generative grammars for natural languages should be structured) originates in the mathematicization of logical proof by the great Polish-born American mathematical logician Emil Leon Post (1897–1954).

3.1 Proof theory and production systems

In his 1920 doctoral dissertation, published as 1921, Post undertook the task of formalizing the logic assumed by Whitehead and Russell in *Principia Mathematica*, providing a provably correct syntactic proof system for it, and showing that the resultant system was decidable. What he ended up with was the discovery that this program could never succeed: there cannot be a decision procedure for theoremhood. But he created new subfields in the process. In particular, he developed a generative characterization of the recursively enumerable (r. e.) sets, and later laid the foundations of recursive function theory (see Post 1944, on which most of Rogers 1967 can be seen as an extended commentary).

Post formalized logical proof through the use of purely syntactic string manipulations defined by rules that he called “productions”. A production associates a set of string schemata $\{\phi_1, \cdots, \phi_n\}$ with a new string schema ϕ_{n+1}, one that $\phi_1, \cdots, \phi_n$ are said to “produce” (or guarantee the derivability of). Intuitively, if axioms or already derivable strings can be found to match $\{\phi_1, \cdots, \phi_n\}$ by fixing appropriate values of the variables therein, then the string matching ϕ_{n+1}, under the same assignment of values to variables, is thereby guaranteed to be derivable. In the case of an inference rule, instantiations of $\phi_1, \cdots, \phi_n$ are premises and the instantiation of ϕ_{n+1} is a conclusion whose legitimacy or provablity those premises are sufficient to guarantee. But production systems were defined in a way so general that they could cover the formation rules and the definition of the set of axioms as well.

A “canonical production system” was defined as a set of initially given strings over some finite symbol inventory (axioms, or “primitive assertions”), together with a set of productions. The set of outputs of such a system (the set of theorems or “assertions” in the case of a logic) was defined as the smallest set containing all the initial strings plus all and only those other strings over the vocabulary that are obtainable via the productions from some set of strings all of which are themselves obtainable.

Post’s general metaschema for productions (Post 1943: 197) was given in terms of the rather alarming display in (1), which needs a little interpretation.

(1)

$$g_{11}\,P_{i'_1}\,g_{12}\,P_{i'_2}\,\cdots g_{1m_1}\,P_{i'_{m_1}}\,g_{1(m_1+1)}$$
$$g_{21}\,P_{i''_1}\,g_{22}\,P_{i''_2}\,\cdots g_{2m_2}\,P_{i''_{m_2}}\,g_{2(m_2+1)}$$
$$\cdots\cdots\cdots\cdots$$
$$g_{k_1}\,P_{i^{(k)}_1}\,g_{k_2}\,P_{i^{(k)}_2}\,\cdots g_{km_k}\,P_{i^{(k)}_{m_k}}\,g_{k(m_k+1)}$$

produce

$$g_1\,P_{i_1}\,g_2\,P_{i_2}\,\cdots g_m\,P_{i_m}\,g_{m+1}$$

The g_i symbols in this array are metasymbols that in actual productions would be specific strings of symbols. (The subscripts on the g’s above the word “produce” are ordered pairs: the jth occurrence of a g symbol in row k has the subscript kj.) The P_i symbols are very different: they are free variables, the values of which are arbitrary sequences of symbols present in the formulas to which the inference rules are applied. (In the lines above the word “produce”, the jth occurrence of a P variable in row k has the subscript $i^{(k)}_j$; for small values of k, the “(k)” is written as a sequence of k primes. The rows may have different numbers of

P's and g's; the number of P variables in row r is indicated by a number m_r, so the number of g's in that row is $m_r + 1$.) A production does not tamper with the values of the P variables, but merely carries over into the conclusion the values assigned when matching premises to the schema.

P variables are rather like the "back references" of the editing language of the Unix stream editor **sed**. A **sed** command that reduplicates a string that follows the symbol **R** can be written in the form "`s/R\(.*\)/R\1\1/`", where "`\(.*\)`" is a variable over arbitrary strings and `\1`, `\2`, etc., are calls for values of such variables (`\1` calls the value of the first or leftmost variable, `\2` calls the second, and so on). Such a command would correspond to a production with one premise written as "**R** P_1 produces **R** P_1 P_1".

Post stipulates that all the P variables in a conclusion must appear somewhere in the premises whose matching to already-obtained strings permit that conclusion to be obtained. One might well ask how this condition—which I will refer to as No New Variables—could possibly be compatible with any rule similar in effect to the 'Disjunction Introduction' familiar from natural deduction: a rule which permits the inference from α to $\alpha \vee \beta$, where choice of β is arbitrary.[2] But in fact there is no conflict here. No New Variables seems superfluous in the sense that productions introducing new arbitrary variables can always be eliminated in favor of new equivalent ones that comply with the condition. This is because Post explicitly allowed production systems to operate on a larger vocabulary of symbols than the set of symbols appearing in the assertions of the system. That is, the vocabulary for a production system generating a set $L \in \Omega_T^*$ can be a set $\Omega = \Omega_T \cup \Omega_N$ where any symbol in Ω can figure in the operations of the productions but the theorems ("assertions" or generated strings) of the system have to be strings over Ω_T.

For example, in the excellent elementary exposition of production systems given by Brainerd & Landweber (1974, chapter 7: 168–170), such extra symbols are exploited in a formalization of the propositional calculus by letting Ω include not just the terminal symbols $\Omega_T = \{p, 1, \neg, \supset, (,)\}$ (where "$\supset$" is the material implication connective) but also a set of extra symbols to classify strings as propositional variables, well-formed formulae, axioms, and theorems. In particular, if we let **T** have the intuitive interpretation "the following string is a theorem", then Modus Ponens, the only rule of inference assumed in the system they formalize, can be formalized thus:

(2) $\{\mathbf{T}\, P_1, \mathbf{T}\, (P_1 \supset P_2)\}$ produce $\mathbf{T}\, P_2$

(An additional production allows **T** P_1 to produce P_1, so the final outputs of the system are the theorems themselves, shorn of their **T** prefixes.)

Suppose we did want to posit in an axiomatic system an inference rule analogous to the natural deduction rule of Disjunction Introduction, allow a theorem α to produce the theorem $\alpha \vee \beta$ for arbitrary β. Using an extra symbol **F** with the intuitive interpretation "the following string is a well-formed formula" (for formation rules can also be expressed as productions), we could formalize it thus:

(3) $\{\mathbf{T}\, P_1, \mathbf{F}\, P_2\}$ produce $\mathbf{T}\, (P_1 \vee P_2)$

And this formulation complies with No New Variables.

3.2 $[\Sigma, \mathrm{F}]$ grammars as production systems

SS defines "the form of grammar associated with the theory of linguistic structure based upon constituent analysis" thus:

(4) Chomsky's definition of $[\Sigma, \mathrm{F}]$ grammars

> Each such grammar is defined by a finite set Σ of initial strings and a finite set F of "instruction formulas" of the form $X \to Y$ interpreted: "rewrite X as Y." Though X need not be a single symbol, only a single symbol of X can be rewritten in forming Y. [*SS*: 29]

Thereafter he refers to a grammar of this form as a "$[\Sigma, \mathrm{F}]$ grammar". He gives this as an example:

[2] Jeff Pelletier raised this question. Lloyd Humberstone and Alasdair Urquhart helped me answer it.

(5) $\Sigma = \{Z\}$
$F = \{Z \rightarrow ab, Z \rightarrow aZb\}$

The stringset generated by (5) is $\{a^n b^n | n \geq 1\}$. Chomsky adds:

> It is important to observe that in describing this language we have introduced a symbol *Z* which is not contained in the sentences of this language. This is the essential fact about phrase structure which gives it its "abstract" character. [*SS*: 31]

Clearly, (4) is a special case of a production system: Σ is the set of initial strings or primitive assertions, and F is the set of productions. One of the ways in which a $[\Sigma, F]$ grammar is more restricted than production systems in general is that its productions are limited to just one premise. But it appears that otherwise the restrictions are not stringent. The grammar in (5) is an unrepresentative one, since apart from the implicit *P* variables at the beginning and end, the left hand sides contain only a single symbol. Chomsky specifically says this does not have to be the case, but he gives no example at that point of a grammar where in a rule $X \rightarrow Y$ we have $|X| > 1$, so the reader has to work out how things operate in that case.

It is important in this connection to keep in mind that rewriting rules as *SS* presents them apply to specified subparts of strings and keep the rest unchanged. What the rule $Z \rightarrow aZb$ means is that $P_1\ Z\ P_2$ produces $P_1\ aZb\ P_2$. When it applies to a string like *aaZbb*, it does not replace the whole thing by *aZb*; it replaces the part that matches the operative part of the left hand side, namely *Z*, so the output is *aaaZbbb*. When Chomsky says that the left hand side of a rule may consist of more than one symbol but only one symbol may be replaced, he clearly means to allow for rules such as "$xAy \rightarrow xBCy$", which means that $P_1\ xAy\ P_2$ produces $P_1\ xBCy\ P_2$. Here again the parts of the string that are not explicitly changed must carry over unchanged (rather than, say, disappear or randomly mutate), so the fact that more can be specified before the arrow than just the single changed symbol means that we are dealing with is context-sensitive rewriting.

The possibility that rewriting might make a string shorter is not ruled out in (4), so a rule like "$xAy \rightarrow xy$" is allowed in $[\Sigma, F]$ grammars as defined in *SS*. This makes them identical with what Harrison (1978: 17) calls "context-sensitive with erasing" grammars, which can generate any r. e. stringset.[3]

In Post's notation, the rules of $[\Sigma, F]$ grammars would look like this:

(6) $P_1\ g_1\ g_2\ g_3\ P_2$ produces $P_1\ g_1\ g_4\ g_3\ P_2$

The extra symbols that Chomsky describes as essential to the abstract character of phrase structure correspond to the extra symbols in Ω_N that do not appear in assertions. In later work, though not in *SS*, Chomsky refers to these extra symbols as "non-terminals". Davis (1994a, xiv) takes the *P* variables in a production system to correspond to the non-terminals of formal language theory, but that is an error. Non-terminals like NP or V in *SS* are drawn from a fixed, finite inventory of symbols that constitutes the alphabet for the strings that productions manipulate. Post's *P* variables, by contrast, are drawn from an infinite set of indexed variables that form part of the metatheoretical apparatus and which never appear in the symbol strings manipulated by productions.

The terminological habits of formal language theorists may have misled Davis on this point: many computer science texts do refer to non-terminals in CF-PSGs as "variables" because they act in a sense as variables over substrings in the terminal vocabulary: 'NP' can stand for (i.e., have as its terminal yield) *the boy* or *a girl*, etc. But this an entirely different notion from the one captured by Post's *P* variables.

4 Transformations

When transformations are introduced in *SS* they are not defined with any precision. In fact they are not really distinguished from informal descriptions of their effects (the discussion

[3] Chomsky 1956b was more careful, and had the additional stipulation: "Neither the replaced symbol nor the replacing string may be the identity element" (p. 117 in the 1956 version, p. 112 in the 1965 revision). That limits $[\Sigma, F]$ grammars to the context-sensitive.

of the coordination principle in section 4.3 below makes this particularly clear). And some of the statements made about them are clearly false. I will consider three specific examples in the sections that follow.

4.1 Singulary transformations: Affix Hopping

The transformational rule known to many linguists as Affix Hopping, called "the Auxiliary Transformation" in *SS*, is initially given in this form (*SS*: 39):

(7) Let *Af* stand for any of the affixes *past*, *S*, *Ø*, *en*, *ing*. Let *v* stand for any *M* or *V*, or *have* or *be* (i.e., for any non-affix in the phrase *Verb*). Then:

$Af + v \rightarrow v\ Af\ \#$, where # is interpreted as word boundary.

It is claimed (*SS*: 40) that this rule "violates the requirements of $[\Sigma, \mathrm{F}]$ grammars ... severely" in that it "requires reference to constituent structure (i.e., past history of derivation) and in addition, we have no way to express the required inversion within the terms of phrase structure." Neither claim is true.

First, on derivational history, by "reference to constituent structure (i.e., past history of derivation)" Chomsky means that in order to know whether (7) can apply to this string:

(8) *the* + *man* + *S* + *have* + *en* + *be* + *ing* + *read* + *the* + *book*

it is necessary to know that *read* was introduced by a step that had $\phi\ V\ \psi$ as its input line ($\phi, \psi \in \Omega^*$) and rewrote it as ϕ *read* ψ, which establishes that *read* "is a *V*", so (7) can apply.

But Chomsky seems to have failed to appreciate the power that the availability of extra symbols affords him. In order that it should be possible to read off that *read* is a *V*, all that is necessary is to carry over an extra symbol that says so. There is actually no need for a reference specifically to *V* here: *V* acts the same way as *M* or *have* to *be*. What is needed is identification of the items that count as falling under *v* and the items that count as falling under *Af*. This could be done by introducing a new symbol **V** with the intuitive meaning "the immediately following symbol counts as an instance of *v*", and a new symbol **A** with the intuitive meaning "the immediately following symbol counts as an instance of *Af*". Instead of (8) we would have (9).

(9) *the* + *man* + **A** *S* + **V** *have* + **A** *en* + **V** *be* + **A** *ing* + **V** *read* + *the* + *book*

The special symbols could then be eliminated by rules equivalent to productions like "$P_1 + \mathbf{V}P_2 + P_3$ produces $P_1 + P_2 + P_3$" and so on. Rules do not need to be able to function like "a more powerful machine, which can 'look back' to earlier strings in the derivation in order to determine how to produce the next step in the derivation" (*SS*: 38). The engineering metaphor seems entirely misguided.

Second, the claim that "we have no way to express the required inversion within the terms of phrase structure" (to transform "*Af v*" into "*v Af*") is false. By "within the terms of phrase structure" Chomsky means within the terms of $[\Sigma, \mathrm{F}]$ grammars. But as we have seen, these are context-sensitive, not context-free, so they permit rules like "$xAy \rightarrow xBy$" (or in the notation familiar from generative phonology, $A \rightarrow B \,/\, x\_\_y$). Using a sequence of rules of this form it is straightforward to convert a string *AB* into the string *BA*: it can be done by the three rules $AB \rightarrow \gamma B$, $\gamma B \rightarrow \gamma A$, and $\gamma A \rightarrow BA$, where γ is some non-terminal not used elsewhere. The same holds for converting '*ing* + *read*' (or **A** *ing* + **V** *read*) into '*read* + *ing* #'. Only relative to a certain fixed choice of symbol inventory can such an inversion be said to fall outside the range of what $[\Sigma, \mathrm{F}]$ grammars can do.[4]

[4] The first printing of *SS* contained a clearly erroneous statement about the power of $[\Sigma, \mathrm{F}]$ grammars. On page 31 it was claimed that the copying stringset $\{xx | x \in (\mathbf{a}, \mathbf{b})^+\}$ "cannot be produced by a grammar of this type." This is not true. A grammar generating it is given as the solution to an exercise by Partee, ter Meulen and Wall (1993: 631, top). Some time in 1959 or later a correction was made to the plates of *SS*: the words "unless the rules embody contextual restrictions" were added, along with a footnote reference to Chomsky (1959). But as just noted, the rules in a $[\Sigma, \mathrm{F}]$ grammar as defined already incorporate contextual restrictions. The correction should have said that the copying stringset cannot be produced by a grammar like (5) in which the left hand side is a single symbol—a CF-PSG.

It has never been clear to me why so many linguists who encountered *SS* regarded the Auxiliary Transformation and the rest of the analysis of the verb and auxiliary system as elegant and attractive. The analysis suffers from a host of fairly serious problems. It gives rise to various ordering paradoxes and entails various counterintuitive constituency claims (Gazdar, Pullum & Sag 1982, 613–616, provide a brief summary).

As one example of the kind of problem I refer to, consider the rule "$V \to V_1\ Prt$", proposed on p. 75 as a way of treating verb-particle constructions like *bring in* as verbs. This rule is incompatible with the Auxiliary Transformation no matter what we assume about V_1. If we assume $V \neq V_1$, then since V falls under v but V_1 does not, the Auxiliary Transformation will produce **bring inned* and **bring inning* instead of *brought in* and *bringing in*. If we change the definition of v to include V_1, both the desired *brought in* and *bringing in* and the undesired **bring inned* and **bring inning* will be generated. And if we assume $V = V_1$, then we get recursion, leading to **bring in up*, **bring in up out*, **bring in up out on*, etc., as well as incorrect affix placements like **bring inning up* and **bring in upping out*.

Perhaps the most serious theoretical criticism of the *SS* treatment of auxiliaries is the observation made by Sampson (1979): that the analysis is not compatible with the formal theory of Chomsky's magnum opus *The Logical Structure of Linguistic Theory* (1956a, henceforth *LSLT*), of which *SS* is supposed to be in effect an informal digest. Sampson notes that (7) is not a legal transformation under the theory of *LSLT*. The reason has to do with the status of the cover symbols *v* and *Af*. These are neither terminal symbols nor non-terminal symbols; they are ad hoc devices, not sanctioned by the *LSLT* theory, with the function of enabling 16 different transformations that share most of their structure to be (apparently) collapsed into one. Thus one of the most famous of all the transformations in *SS* is not a transformation at all under the *LSLT* definition.

4.2 Transformations with essential variables: *wh*-movement

Nothing so far illustrates any variables in *SS* that are like Post's *P* variables in covering arbitrary and unbounded substrings that are not necessarily constituents; but *SS* does include one such case. The *wh*-fronting transformation[5] called "T_{W_1}" is formally stated thus (*SS*: 112):

(10) T_{W_1}: Structural analysis: $X - NP - Y$ (X or Y may be null)
Structural change: $X_1 - X_2 - X_3 \to X_2 - X_1 - X_3$

What the rule does is in effect to shift an NP to the beginning of a sentence across an unbounded domain that may contain any arbitrary sequence of symbols, which illustrates very clearly the difference between non-terminal symbols and Post-style *P* variables. In later transformational literature, e.g. Postal (1971), variables of this sort became known as "essential variables". Ross's celebrated dissertation (1967) is an exploration of universal constraints that might be placed on rules making reference to them.

The T_{W_1} rule is tagged "optional and conditional on T_q" (*SS*: 112), where T_q is the rule usually known as Subject-Auxiliary Inversion, which shifts to the left of an NP (i) an immediately following concord morpheme (*S* or *Ø* or *past*) plus any instance of *M* or *have* or *be* that may immediately follow that. The idea seems to be to front an NP only if the sentence begins with an element falling under *v*. But exactly what is meant by making one rule "conditional on" another is not explained: it would appear to be a global constraint on derivations of the sort that in later work Chomsky would resolutely oppose.

The notational practice seen in (10), and in the formulation of all rules in the list in pp. 112–114, is never explained in *SS*, and seems quite odd. There is no connection at all between the X in the structural analysis (the input description) and the various X_i in the structural change (the output description). And although nothing is explicitly said, it is apparent that for each $i \geq 1$ we are supposed to match up X_i to the i^{th} element in the structural analysis. It is strange that five variables are used to hold three values in (10). In Post's notation, the formulation would be simpler, with only two variables:

(11) $P_1\ NP\ P_2$ produces $NP\ P_1\ P_2$

[5] It actually fronts NP, but there is a second subrule called T_{W_2} which changes *NP* at the beginning of a sentence into *wh* + *NP*, and morphophonemic rules turn this into *who* or *what*.

4.3 Generalized transformations: Conjunction

SS also uses rules that would correspond to productions with more than one premise: these are his “generalized transformations”. Six years after the publication of *SS* Chomsky & Miller (1963: 284) proposed that all rules of grammar can be given in this form:

(12) $\phi_1, \ldots, \phi_n \rightarrow \phi_{n+1}$

They explain: “each of the ϕ_i is a structure of some sort and ... the relation $\rightarrow$ is to be interpreted as expressing the fact that if our process of recursive specification generates the structures $\phi_1, \ldots, \phi_n$ then it also generates the structure ϕ_{n+1}.” It should be clear that this just summarizes Post’s production systems, omitting the details relating to the free variables and the process of assigning them their values. The generalized transformations in *SS* are the ones where $n = 2$.

The first such rule that Chomsky considers is called “Conjunction” on p. 113, but it is introduced informally on p. 36, where it is stated thus:

(13) If S_1 and S_2 are grammatical sentences, and S_1 differs from S_2 only in that X appears in S_1 where Y appears in S_2 (i.e., $S_1 = \ldots X \ldots$ and $S_2 = \ldots Y \ldots$), and X and Y are constituents of the same type in S_1 and S_2, respectively, then S_3 is a sentence, where S_3 is the result of replacing X by $X + \mathit{and} + Y$ in S_1 (i.e., $S_3 = .. X + \mathit{and} + Y ..$).

This is referred to as a “rule”, but it is not a transformation in any formal sense. S_1 and S_2 are required here to be ‘grammatical sentences’; i.e., strings actually generated by the grammar. So (13) involves existential quantification over the entire content of the language. It is what would later be called a transderivational constraint: the grammaticality of S_3 depends on the independently assessed grammaticality of two other sentences, S_1 and S_2.

Note in passing that the claim expressed by (13) is not true of English. There are many cases of X and Y such that both can occur in a given context but the coordination *X and Y* cannot. An obvious one involves verb agreement. If X = *Xavier* and Y = *Yves*, for *I think X is eligible* and *I think Y is eligible* we have the prediction from (13) that **I think <u>Xavier and Yves</u> is eligible* should be grammatical, but it is not. Several other such failures of (13) are catalogued by Huddleston & Pullum (2002, pp. 1323–1326).

There is no attempt in *SS* to deal with multiple coordination—cases where there are more than two coordinates. No kind of finite production system can provide the kind of analysis for multiple coordination that seems linguistically desirable (namely, an analysis with unranked trees with no bound on branching degree), because in any such system there is a longest right hand side. Generalized transformations are likewise of no use: an infinite set of rules of the form in (12) would be called for, one for each *n*. The problem lingers down to recent times; see Borsley 2003 for a critique of contemporary work that tries to solve the problem by reducing all coordination to binary structure. (Note, however, that a beautiful solution is available in non-generative terms: see Rogers 1999.)

Chomsky recognizes that “additional qualification is necessary” to his description, but nonetheless claims that “the grammar is enormously simplified if we set up constituents in such a way that [(13)] holds even approximately” (*SS*: 37). So let us consider just the matter of formulation.

What is really striking is that when (13) is stated more formally in the summary rules at the end of the book (p. 113), the result is considerably less explicit and less accurate than (13). The rule statement is given in (14).

(14) Structural analysis: of S_1: $Z - X - W$
of S_2: $Z - X - W$
where X is a minimal element (e.g., *NP*, *VP*, etc.) and
Z, W are segments of terminal strings.
Structural change: $(X_1 - X_2 - X_3; X_4 - X_5 - X_6) \rightarrow$
$X_1 - X_2 + \mathit{and} + X_5 - X_3$

It is now clear that S_1 and S_2 will not be sentences (strings over the terminal vocabulary); they will be stages in a derivation, including nonterminals.

X is stipulated to be a ‘minimal element’, and although this term is undefined, it appears to mean ‘single nonterminal’. Z and W, however, are stipulated in a prose annotation to be 'segments of terminal strings'. This means, assuming that the string values are supposed to be the same in each case, that S_1 and S_2 are completely identical, and there was no point in distinguishing them. They will be what Lasnik (2000: 31) calls "monostrings": strings containing only one non-terminal. In the case where $X = NP$, they will be strings like *Put* NP *in the truck*. The intent is that since *Put the dog in the truck* and *Put the toolbox in the truck* are both grammatical, and *the dog* and *the toolbox* can both be the terminal yield of an NP, *Put the dog and the toolbox in the truck* should also be grammatical.

A small problem immediately becomes apparent: (14) does not guarantee any difference between the terminal strings of the X constituents in S_1 and S_2, so (14) yields **Put it and it in the truck* as an output, which is probably unintended (since in (13) it was stated that 'S_1 differs from S_2': the intention was for S_1 and S_2 to be identical sentential forms that are somehow guaranteed to have distinct generated terminal strings).

Closer examination reveals that nothing really turns on making reference to S_1 and S_2 at all. No use is ever made of the variables Z and W in the 'structural change'. We are apparently supposed to intuit that (i) all of the X_i variables range over terminal strings; (ii) $X_1 = X_4 = Z$; (iii) $X_3 = X_6 = W$; (iv) $X_2 \neq X_4$; and (v) X_2 and X_4 are terminal strings of instances of the category X. None of this is made explicit in (14) or elsewhere.

The "Conjunction" transformation seems to be a remarkably inexpert use of mathematical symbolism, but for what it is worth, it would be straightforwardly expressible as a production with two premises in Post's formalism. However, its descriptive content appears to be specifiable much more simply. All it really does is to ensure that a nonterminal symbol X can exhaustively dominate a string of the form '*X and X*', in any context where X can appear. CF-PSG rules could do that. And a rule like '$NP \rightarrow NP$ *and* NP' would represent *the dog and the toolbox* as an NP, which (14) does not do. (See Gazdar 1981, which begins with essentially that observation and derives from it some remarkably wide-ranging conclusions).

5 Generative capacity

There seems to be no important difference in mathematical character between a TGG and a production system. In saying this I do not mean to deny that certain specific organizational proposals are explicit or implicit in *SS*. For example, Chomsky bifurcates the grammar into a set of non-recursive CF-PSG productions generating a finite "kernel" and a set of transformations providing derivations for the rest of the sentences. But one could equally well take a production system defining the propositional calculus and set it out with formation rules like "$\mathbf{F}\,P$ produces $\mathbf{F}\,(\neg P)$" segregated from the transformation rules like Modus Ponens (as (2) formalizes it). Such a separation would seem to be a presentational decision about formalisms, not a substantive claim about languages or logics. Production systems are generative grammars of an extremely general sort, and all the rules of *SS* seem to fall into place within the theory of Post's production systems.

Chomsky does add some elaborations to production systems that I have not yet mentioned. One is "extrinsic" rule ordering: the requirement that a grammar should define a strict total order on its rules, each rule being permitted to apply only if it is applicable to what has been obtained after all the rules ordered before it have had their chance to apply, and before any of the rules ordered after it have had their chance. Another is the classification of rules into the optional (which are permitted to apply when their turn in the ordering comes but do not have to) and the obligatory (which must apply if they can when their turn comes). But these devices do not seem to introduce any new mathematical possibilities. No one ever offered an example of a stringset that can be generated by some unordered set of productions but cannot be generated by any ordered set (even over a vocabulary containing additional non-terminals).[6] Likewise no one ever offered a case in which tagging rules as optional or obligatory was an absolute necessity. Productions are intuitively optional: where more than one production is applicable to some substring, any of the eligible ones is permit-

[6] This is different from saying that ordering cannot restrict what a particular set of rules can generate. Pelletier (1980) shows that requiring strict ordering of a set of rules can indeed make some outputs impossible to generate. But as he stresses, this result presumes that the set of rules is fixed, which is not the position linguists are ever in.

ted to apply. But all that is necessary to make a rule *r* intuitively obligatory is to ensure the presence in the relevant strings of some non-terminal that only *r* can eliminate.

Considerations such as these relate to the question of the generative capacity or expressive power of grammars. And it turns out that Post, having in effect invented generative grammars, also proved the first two theorems concerning generative capacity. He called his maximally general production systems, with productions as in (1), “canonical systems”, and his major result in Post (1943) was a theorem concerning the expressive power of production systems having a radically limited format for productions.

5.1 Normal systems

Post (1943) proves that every set generated by a canonical system can also be generated by a system in a much more restricted format called a “normal system”. In a normal system there is just one axiom, and all productions have a single premise and take this form, where *x* and *y* are particular given strings:

(15) xP produces Py

To be more precise, Post’s theorem is stated as in (16):

(16) **Theorem** (Post 1943) Given a canonical system Γ over a finite vocabulary Ω_T it is always possible to construct a normal system Γ' over $\Omega = \Omega_T \cup \Omega_N$ such that $x \in \Omega_T^*$ is an assertion of Γ' iff *x* is an assertion of Γ.

Thus even a radical restriction on rule form, limiting all rules to saying “A string beginning with *x* may be rewritten with its *x* prefix erased and *y* added at the end”, may have no effect at all on generative capacity.

5.2 Semi-Thue rules

There is another specially limited form of productions. Chomsky (1962) calls these “rewriting rules”, and recognizes explicitly that they are restricted forms of Post’s production systems:

> A rewriting rule is a special case of a production in the sense of Post; a rule of the form ZXW $\rightarrow$ ZYW, where Z or W (or both) may be null. (Chomsky 1962: 539)

This is just another way of presenting the $[\Sigma, \mathrm{F}]$ grammars considered earlier; in Post’s notation the rules would have the form shown in (6). But the particular special case under consideration originates in a technical paper from ten years before in which Post (following a suggestion by Alonzo Church) tackled an open question posed by Axel Thue (1914). Thue had asked whether there was a decision procedure for determining whether a specified string X could be converted into a given string Y by a set of rules of the form “$WXZ \leftrightarrow WYZ$”, where W, X, Y, Z are strings over some fixed finite alphabet and $\phi \leftrightarrow \psi$ is to be read as “ϕ may be replaced by ψ or conversely”.

The problem might be seen as motivated by logical equivalences such as DeMorgan’s Laws $(\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q)$ and $(\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q))$: using some finite set of logical equivalencies such as these, is the formula φ_1 logically equivalent to the formula φ_2 or not?

Post (1947) answers Thue’s question by reduction: he proves that (i) if we could decide derivability for a Thue system (where for any rule $\phi \rightarrow \psi$ belonging to the system the inverse $\psi \rightarrow \phi$ also belongs) we could also decide it for “semi-Thue” systems where the inverse is not present; and (ii) that would mean decidability of the full range of stringsets that normal or canonical systems can generate, which is provably unsolvable. “Semi-Thue” productions are of course simply Chomsky’s type-0 rules.

The actual systems studied by Thue are not generative grammars: they have no initial strings and no distinction between terminal and non-terminal symbols. But Post’s interest was in what rules of the semi-Thue form could generate, and the theorem he proved has direct application to generative grammars.

5.3 Recursive enumerability

Post understood that the class of stringsets that canonical production systems can generate is the r. e. stringsets. Not only do arbitary systems of rules of inference as formalized by canonical systems yield all and only the r. e. stringsets, but the same is true for rules in the much more limited normal systems and semi-Thue systems. Both results show that radical limitations on the form of rules may have no effect on what can be generated.

The importance of Chomsky (1959) lay in its demonstration that other restrictions did limit what could be generated. If erasing is forbidden, context-sensitive grammars generate only the context-sensitive ("type-1") stringsets; and rules with only one symbol on the left hand side generate only the context-free ("type-2"). But the transformations introduced in *SS* observed no such restrictions.

Hilary Putnam (1961) saw the implications of this very clearly. After discussing some intuitive reasons why we should want to regard natural language stringsets as recursive (decidable) sets, he stated (p. 41): "Chomsky's general characterization of a transformational grammar is much too wide. It is easy to show that any recursively enumerable set of sentences could be generated by a transformational grammar in Chomsky's sense." He gave no proof of this, but his conclusion was correct, as others later verified in detail (see Peters & Ritchie 1971 and 1973, inter alia). And clearly, if even normal systems (where rules can say nothing more than "g_1 P produces P g_2") capture the r. e. stringsets, it is hard to see how the rule formalism of *SS* could not. After all, $[\Sigma, \mathrm{F}]$ grammars already generate any r. e. set, and *SS* attempts to show that those are not expressive enough.

6 Conclusions

This paper has concerned itself only with some of the mathematical and logical foundations and antecedents of *SS*, and the coherence of its formalism.[7] One might ask why we should care about faults in a monograph that is now more than 50 years old. The answer I would give is that myths about scientific breakthroughs and results can warp practitioners' perceptions of the history of a field, with bad consequences for the conduct of science.

We know from the history of science in general that it is often wrong to attribute a new idea to one person. There is no answer to whether Wallace or Darwin conceived of evolution by natural selection, whether Priestley or Lavoisier discovered oxygen, whether the calculus is due to Leibniz or Newton (or even Cauchy, who first provided it with a rigorous mathematical basis). Discoveries and innovations develop over time and build on earlier developments and adjacent fields. A monogenesis myth that has a research program springing from nowhere in the mind of a lone genius may be bad for science in at least two ways, both of which are worth guarding against.

The first is that such myths encourage linguists in complacent maintenance of false assumptions. If almost everyone believes that *SS* showed transformations to be necessary back in 1957, non-transformational syntactic research is bound to remain underdeveloped and underexplored, as indeed I think it has been over the past fifty years.

The second is that they promote biased and lazy citation practices: passing the same old references from paper to paper without anyone checking that the sources said what people think they said. *SS* does not properly credit the earlier literature on which it draws. Chomsky has never cited any paper of Post's other than Post (1944), an informal paper deriving from a lecture to the American Mathematical Society on r. e. sets of positive integers, which is cited in Chomsky (1959: 137n) and Chomsky (1961: 7) as an example of someone using the term "generate" in the mathematical sense. Post is mentioned without a bibliographical citation in two other places (Chomsky 1962: 539, and 1965: 9), but Chomsky has never cited Post's

[7] The empirical claims *SS* makes about English are also thoroughly flawed, but I do not have space to discuss them here. Note also that Harman (1963) published a response to *SS*, showing that through the use of what were in effect feature distinctions on category labels a CF-PSG could do all of what transformations were called upon to do in *SS*. Gazdar (1982: 134) provides a retrospective appreciation of Harman's work and a criticism of the "terminological imperialism" of Chomsky's unsatisfying response to it.

technical papers on production systems.[8] Other linguists, and even historians of linguistics (the generally very interesting work of Tomalin 2006, for example), tend to simply follow Chomsky, citing only what he cites.

The contributions of Zellig Harris are also somewhat downplayed in *SS*. The standard view is that Harris worked solely on methods of "taxonomic" bottom-up utterance analysis and *SS* introduced grammars that synthesized sentences top-down. But Harris (as Seuren 2009 points out) explicitly envisages top-down generation:

> The work of analysis leads right up to the statements which enable anyone to synthesize or predict utterances in the language. These statements form a deductive system with axiomatically defined initial elements and with theorems concerning the relations among them. The final theorems would indicate the structure of the utterances of the language in terms of the preceding parts of the system. [Harris 1951: 372–373, quoted by Seuren 2009: 107]

So Harris clearly saw that formal axiomatic systems could be exploited as generative production systems, generating well-formed strings rather than logical theorems. And he saw it ten years before *SS* appeared[9] (the preface to *Methods in Structural Linguistics* is dated January 1947, and credits the young Chomsky, who read it in proof when he was an undergraduate, for "much-needed assistance with the manuscript").

SS is credited with a degree of originality, explicitness, and technical coherence that it does not actually exhibit, but to say that is not to deny that somehow it managed to stimulate other linguists to strive for these virtues. Its effect was catalytic rather than substantive (it contains no results that are defended in detail today). It may be that some will dismiss the foregoing discussion as just a negative book review offered fifty years too late, but in a sense that would underrate the importance of *SS*. Only in the light of the subsequent developments in linguistics that *SS* managed to encourage could my evaluation of its content have been undertaken. It would have been very useful for linguists to have access, by about 1960, to a detailed critical review of *SS*; but the simple fact is that it would have been impossible, because absolutely no linguist at that time could have written it.

Acknowledgements My philosopher collaborator Barbara Scholz contributed generously to this research, helping to develop the ideas as well as advising on how better to express them. I thank her and Gerald Gazdar, Lloyd Humberstone, Robert Levine, Jeff Pelletier, Geoffrey Sampson, and Thomas Wasow for helpful criticisms on earlier drafts; Frederick Newmeyer, Pieter Seuren, Marcus Tomalin, and Alasdair Urquhart for useful conversations and correspondence; my referees for their input; and Andrew Garrett for kindly providing me with a copy of the first printing of *SS*. These people do not necessarily agree with what I have decided to say (some definitely don't), so they should not be blamed for my errors.

References

Bever TG, Fodor JA, Garrett M (1968) A formal limitation of associationism. In: Dixon TR, Horton DL (eds) Verbal Behavior and General Behavior Theory, Prentice-Hall, Englewood Cliffs, NJ, pp 582–585

Borsley RD (2003) Against ConjP. Lingua 115:461–482

Brainerd WS, Landweber LH (1974) Theory of Computation. John Wiley, New York

Carnap R (1934) Logische Syntax der Sprache. Julius Springer, Vienna, translated as *The Logical Syntax of Language*, Kegan Paul, 1937

Chomsky N (1956a) The Logical Structure of Linguistic Theory. MIT Library, Cambridge, MA, microfilmed; revised version of a 1955 unpublished manuscript

Chomsky N (1956b) Three models for the description of language. IRE Transactions on Information Theory IT-2:113–123, reprinted with substantive revisions in Luce, Bush & Galanter (1965), 105–124

Chomsky N (1957) Syntactic Structures. Mouton, The Hague

[8] Urquhart (2009) suggests that this might be because Chomsky's understanding of Post systems came from a secondary source, namely Rosenbloom (1950), which is cited in Chomsky (1956a) and Chomsky (1956b). *SS* cites neither Rosenbloom (1950) nor anything by Post.

[9] Newmeyer is mistaken in stating (1986: 5n) that the generative proposals Chomsky made in his undergraduate and master's theses on Hebrew in 1949 and 1951 antedate Harris's insights.

Chomsky N (1959) On certain formal properties of grammars. Information and Control 2:137–167, reprinted in Luce, Bush & Galanter (1965, 125–155; citation to original is incorrect)

Chomsky N (1961) On the notion 'rule of grammar'. In: Proceedings of the Twelfth Symposium in Applied Mathematics, American Mathematical Society, Providence, RI, pp 6–24, reprinted with slight revision in Jerry A. Fodor and Jerrold J. Katz (eds.), The Structure of Language: Readings in the Philosophy of Language, 155–210 (Englewood Cliffs, NJ: Prentice-Hall)

Chomsky N (1962) Explanatory models in linguistics. In: Nagel E, Suppes P, Tarski A (eds) Logic, Methodology and Philosophy of Science: Proceedings of the 1960 International Congress, Stanford University Press, Stanford, CA, pp 528–550

Chomsky N (1965) Aspects of the Theory of Syntax. MIT Press, Cambridge, MA

Chomsky N (1975) The Logical Structure of Linguistic Theory. Plenum, New York, published version of Chomsky (1956a)

Chomsky N, Miller GA (1963) Introduction to the formal analysis of natural languages. In: Luce RD, Bush RR, Galanter E (eds) Handbook of Mathematical Psychology, vol II, John Wiley and Sons, New York, pp 269–321

Daly RT (1974) Applications of the Mathematical Theory of Linguistics. Mouton, The Hague

Davis M (1994a) Emil L. Post: His life and work. In Davis 1994b, xi–xxviii

Davis M (ed) (1994b) Solvability, Provability, Definability: The Collected Works of Emil L. Post. Birkhäuser, Boston

Firth JR (1957) Papers in Linguistics 1934–1951. Oxford University Press, London

Fitch WT, Hauser MD (2004) Computational constraints on syntactic processing in nonhuman primates. Science 303:377–380

Gazdar G (1981) Unbounded dependencies and coordinate structure. Linguistic Inquiry 12:155–184

Gazdar G (1982) Phrase structure grammar. In: Jacobson P, Pullum GK (eds) The Nature of Syntactic Representation, D. Reidel, Dordrecht, Netherlands, pp 131–186

Gazdar G, Pullum GK, Sag IA (1982) Auxiliaries and related phenomena in a restrictive theory of grammar. Language 58:591–638

Harman G (1963) Generative grammars without transformation rules. Language 39:597–616

Harris ZS (1951) Methods in Structural Linguistics. Oxford University Press, New York

Harrison M (1978) Introduction to Formal Language Theory. Addison-Wesley, Reading, MA

Huddleston R, Pullum GK (2002) The Cambridge Grammar of the English Language. Cambridge University Press, Cambridge

Joos M (ed) (1957) Readings in Linguistics I: The Development of Descriptive Linguistics in America since 1925, 1st edn. American Council of Learned Societies, Washington, DC

Lasnik H (2000) Syntactic Structures Revisited: Contemporary Lectures on Classic Transformational Theory. MIT Press, Cambridge, MA

Lees RB (1957) Review of Noam Chomsky, Syntactic Structures. Language 33:375–408

Lightfoot D (2002) Introduction. In Noam Chomsky, Syntactic Structures, second edition, v–xviii. Mouton de Gruyter, Berlin

Luce RD, Bush RR, Galanter E (eds) (1965) Readings in Mathematical Psychology, vol II. John Wiley & Sons, New York

Lyons J (1970) Chomsky. Fontana, London

Newmeyer FJ (1986) Has there been a 'Chomskyan revolution' in linguistics? Language 62(1):1–18

Partee BH, ter Meulen A, Wall RE (1993) Mathematical Methods in Linguistics, corrected first edn. Kluwer Academic, Dordrecht

Pelletier FJ (1980) The generative power of rule orderings in formal grammars. Linguistics 18(1/2 (227/228)):17–72

Pereira F (2000) Formal grammar and information theory: Together again? Philosophical Transactions of the Royal Society 358(1769):1239–1253

Peters PS, Ritchie RW (1971) On restricting the base component of transformational grammars. Information and Control 18:483–501

Peters PS, Ritchie RW (1973) On the generative power of transformational grammars. Information Sciences 6:49–83

Philological Society (1957) Studies in Linguistic Analysis. Philological Society, Oxford

Post E (1921) Introduction to a general theory of elementary propositions. American Journal of Mathematics 43:163–185, reprinted in Jan van Heijenoort, ed., From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931, Harvard University Press, Cambridge MA, 1967, 264-283

Post E (1943) Formal reductions of the general combinatory decision problem. American Journal of Mathematics 65:197–215, reprinted in Davis 1994b, 442–460.

Post E (1944) Recursively enumerable sets of positive integers and their decision problems. Bulletin of the American Mathematical Society 50:284–316, reprinted in Davis 1994b, 461–494.

Post E (1947) Recursive unsolvability of a problem of Thue. Journal of Symbolic Logic 12:1–11, reprinted in Davis 1994b, 503–513.

Postal PM (1971) Crossover Phenomena. Holt Rinehart and Winston, New York

Pullum GK, Scholz BC (2007) Systematicity and natural language syntax. Croatian Journal of Philosophy 7:375–402

Pullum GK, Scholz BC (2009) For universals (but not finite-state learning), visit the zoo. Behavioral and Brain Sciences 32(5):466–467

Putnam H (1961) Some issues in the theory of grammar. In: Jakobson R (ed) Structure of Language and Its Mathematical Aspects, no. 12 in Proceedings of Symposia in Applied Mathematics, American Mathematical Society, Providence, RI, pp 25–42

Rogers J (1999) The descriptive complexity of generalized local sets. In: Kolb HP, Mönnich U (eds) The Mathematics of Syntactic Structure: Trees and their Logics, no. 44 in Studies in Generative Grammar, Mouton de Gruyter, Berlin, pp 21–40

Rogers J, Pullum GK (2010) Aural pattern recognition experiments and the subregular hierarchy. In this issue

Rogers, Jr H (1967) The Theory of Recursive Functions and Effective Computability. McGraw-Hill, New York

Rosenbloom P (1950) The Elements of Mathematical Logic. Dover, New York

Ross JR (1967) Constraints on variables in syntax. PhD thesis, MIT, Cambridge, MA, duplicated version published in 1968 by Indiana University Linguistics Club, Bloomington, IN. Later published in book form as Infinite Syntax! (Norwood, NJ: Ablex, 1986)

Sampson G (1979) What was transformational grammar? Lingua 48:355–378, reprinted in Empirical Linguistics, Continuum, 2001

Scholz BC, Pullum GK (2007) Tracking the origins of transformational generative grammar. Journal of Linguistics 43:701–723

Seuren P (2009) Concerning the roots of transformational generative grammar. Historiographia Linguistica 36(1):97–115

Skinner BF (1957) Verbal Behavior. Appleton-Century-Crofts, New York

Svenonius L (1957) Review of 'Three models for the description of language' by Noam Chomsky. Journal of Symbolic Logic 23:71–72

Thue A (1914) Probleme über Veränderungen von Zeichenreihen nach gegebenen Regeln. In: Skrifter utgit av Videnskapsselskapet i Kristiana, I, no. 10 in Matematisk-naturvidenskabelig klasse, Norske Videnskaps-Akademi, Oslo

Tomalin M (2006) Linguistics and the Formal Sciences. Cambridge University Press, Cambridge

Urquhart A (2009) Emil Post. In: Gabbay DM, Woods J (eds) Handbook of the History of Logic, Volume 5: Logic from Russell to Church, North-Holland, Amsterdam, pp 617–666