ANNUAL REVIEWS

Annual Review of Linguistics

Three Mathematical Foundations for Syntax

Edward P. Stabler[1,2]

[1]Department of Linguistics, University of California, Los Angeles, California 90095, USA; email: stabler@ucla.edu

[2]Nuance Communications, Sunnyvale, California 94085, USA; email: edward.stabler@nuance.com

Annu. Rev. Linguist. 2019. 5:243–60

First published as a Review in Advance on September 5, 2018

The *Annual Review of Linguistics* is online at linguist.annualreviews.org

https://doi.org/10.1146/annurev-linguistics-011415-040658

Keywords

syntax, constituency, dependency, Optimality Theory, mathematical foundations

Abstract

Three different foundational ideas can be identified in recent syntactic theory: structure from substitution classes, structure from dependencies among heads, and structure as the result of optimizing preferences. As formulated in this review, it is easy to see that these three ideas are completely independent. Each has a different mathematical foundation, each suggests a different natural connection to meaning, and each implies something different about how language acquisition could work. Since they are all well supported by the evidence, these three ideas are found in various mixtures in the prominent syntactic traditions. From this perspective, if syntax springs fundamentally from a single basic human ability, it is an ability that exploits a coincidence of a number of very different things.

1. INTRODUCTION

The traditional mathematical foundation of syntax assumes that syntax is a theory about which sequences of words are grammatical. A language is regarded as a set of grammatical word sequences, and these sets can be ranked in the Chomsky hierarchy, according to the kinds of grammar that can generate them. A glance at any standard syntax textbook suggests that this idea does not seem to fit the subject. First, while word sequences and grammaticality judgments are relevant, most syntax texts pay considerable attention to trees and other more elaborate structures, where the leaves of those structures are often not words but something more abstract. Furthermore, the focus of study is not tests of grammaticality or any such thing, but is more often comparisons among sentences that show interesting or subtle differences in meaning or acceptability. Second, it does not take much reflection to notice that, if utterances are "grammatical" whenever they can be assigned a linguistic structure by a linguistically competent human in some context (abstracting away from time and memory limitations), then all or almost all sequences of words are certainly grammatical (Manaster-Ramer 1983). Instead, we might count an utterance "grammatical" only if it is actually a "sentence" or "clause," in some specific sense of those terms. But this raises issues about what that sense of "clause" is, and whether, for example, fragment answers to questions are clausal in the relevant sense (Merchant et al. 2013, Weir 2014), since the answer to a question can be any sequence of words:

Q: 'What should I say?'

A: 'Mat the on is cat the.'

That answer is odd but perfectly intelligible; it is a grammatical noun phrase of English, a quotation name of a word sequence, and perhaps clausal in some sense. Every human language allows direct quotation, so in every language, every sequence of words has a grammatical structure. The fact that such issues have not been at all prominent in syntactic theory suggests that the traditional string-based foundation for syntax is missing the mark. A third reason to doubt that syntax is primarily the study of word sequences is that many syntacticians explicitly describe their interest quite differently, sometimes saying, for example, that their primary goal is to describe the relation between form and meaning in such a way that we can explain the acquisition and use of that relation by competent human language users.

This review begins with the idea that a mathematical foundation of syntax should address the question: What are the syntactic structures of human languages? Usually they are trees, or tree-like, so what is the fundamental reason for that? Why does human language have such structures? The first surprise is that answering these questions explains why we should still be interested in the traditional string-based foundations of syntax, even if it is only part of a bigger picture. The second surprise is that the literature presents not one but (at least) three answers to our fundamental questions, three quite different ideas about the fundamental sources of syntax. In fact, in the conservative formulations adopted here, all three answers seem to be correct, and we find them in various mixtures throughout the field. The review concludes by raising the question of how to explain this coincidence.

2. TREES FROM SUBSTITUTION CLASSES

Most introductions to syntax begin, more or less explicitly, with proposals about how to use substitution tests of various kinds to identify categories of words and phrases, comparing grammatical words and sentences with similar sequences that are not grammatical (Adger 2003, Sag et al. 2003, Radford 2009, Sportiche et al. 2014, Bresnan et al. 2015). These tests assume that it makes sense to ask whether a sequence of words s is grammatical, that is, whether it is a grammatical sentence of the language we are studying. Idealizing, we can regard the grammatical sentences of any person's language as a set, an ideally completed, infinite corpus of all the utterances that the person would assign a sentence structure to, abstracting away from limitations of time, memory, and attention.

Let us start with some very simple ideas. Let s be any string, any sequence of words, and let $C(s)$ be the contexts of s in the string language, the set of pairs of word sequences (l, r) such that lsr is a grammatical sentence. Then, let us say that strings s and t are just in case they have exactly the same contexts: $C(s) = C(t)$. Note that with these definitions, for every sentence s, $C(s)$ includes the context that is the pair of empty sequences (ϵ, ϵ), since $\epsilon s \epsilon = s$. Finally, for any string s, let the substitution class $[s]$ be the set of strings intersubstitutable with s.

Starting out in this way, we have not been precise about what is meant by a "word." Is the name *Bill* a word, and if so, is it the same word as the homophonous noun or verb? Is the word *bank* 'riverside' the same as the word *bank* 'financial institution'? Is *bankroll* one word, or two or more? Fortunately, for present purposes, we do not need to answer any of these questions. We can also set aside the problems of word segmentation and identification. We assume only that each utterance presents a finite sequence of words. And while the vocabulary of each speaker may be growing and changing, we assume that it is finite at any particular point in time.

It follows from our definitions that for any word w, $w \in [w]$, and if $C(x) = C(w)$, then $x \in [w]$. So, for example, in English, if *Susan* and *Rebecca* are intersubstitutable in the strong sense we have defined, then *Susan* $\in$ [*Susan*] and *Rebecca* $\in$ [*Susan*].

Given any sets x and y, let xy be the set of strings formed by concatenating each element of x with each element of y. It follows from our definitions that for any strings s and t, $[s][t] \subseteq [st]$. That is, the concatenation of the strings equivalent to s with the strings equivalent to t is always a subset of the strings equivalent to st. Consider, for example, the string language L that has four words, a, b, c, and d, and three sentences, aa, ab, and ac. Then, $[a] = \{a\}$, $[b] = [c] = \{b, c\}$, $[a][b] = \{ab, ac\}$, $[a][a] = \{aa\}$, and $[ab] = \{aa, ab, ac\}$.

A fundamental result in mathematical linguistics, the Myhill–Nerode theorem (see the sidebar titled The Myhill–Nerode Theorem for String Languages), entails that a string language is regular in the sense of being computable by a simple finite automaton if and only if the number of context classes $C(x)$ for substrings x is finite. Let RLs be the set of regular string languages, all the string languages recognizable by finite-state devices.

THE MYHILL–NERODE THEOREM FOR STRING LANGUAGES

Define a *prefix* of string s to be any string t such that, for some u, $tu = s$. For example, the string *Rebecca laughs* has three prefixes: ϵ, *Rebecca*, and the string itself. Given a set of strings L and any prefix s of any string in L, define the right contexts of s: $R(s) = \{t | st \in L\}$. In English, R(*Rebecca*) includes *laughs*, and also *Smith laughs loudly*, and infinitely many other strings. And R(*Rebecca*) $=$ R(*Susan*). Myhill and Nerode show that the string language L is regular (i.e., accepted by a simple finite automaton) if and only if the set of right contexts of prefixes of strings of L is finite. This basic result about how strings relate to grammars has a simple, constructive proof (see, e.g., Hopcroft & Ullman 1969, theorem 3.1, or Salomaa 1973, theorem 5.4; for a fuller, more technical treatment, see Sakarovitch 2009, section 3.3).

Consider, for example, the string language containing odd-length strings of only one word: $L1 = \{a, aaa, aaaaa, \ldots\}$. In this case, note that $R(\epsilon) = R(aa) = R(aaaa) = \ldots$, and $R(a) = R(aaa) = R(aaaaa) = \ldots$. Since $R(\epsilon)$ and $R(a)$ are the right contexts for every prefix of sentences of $L1$, it follows that $L1$ can be accepted by a simple automaton with only two states; therefore, the language is regular.

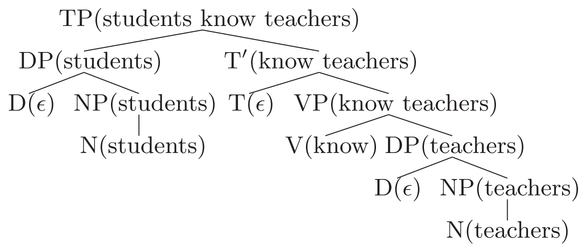
2.1. Phrase Structure Grammars

Chomsky (2006, p. 172n) suggests that, in the work of Zellig Harris and others, phrase structure grammars were designed specifically to express relations between substitution classes. But the deep connection between substitution classes and phrase structure grammars has not been clear until fairly recently. This section begins with a review of these basics.

Context-free grammars (CFGs) are the most commonly used phrase structure grammars, and we presume that readers have encountered them before (see, e.g., Hopcroft & Ullman 1969, section 4). To set the stage for later developments, let us write CFG rules in the logic-based form shown below, using TP in this example as the basic category for sentences:

$$
\begin{array}{ll}
\mathrm{TP}(st) \leftarrow \mathrm{DP}(s)\ \mathrm{T}'(t) & \mathrm{DP}(st) \leftarrow \mathrm{D}(s)\ \mathrm{NP}(t) \\
\mathrm{T}'(st) \leftarrow \mathrm{T}(s)\ \mathrm{VP}(t) & \mathrm{D}(\text{some}) \leftarrow \\
\mathrm{T}(\text{will}) \leftarrow & \mathrm{D}(\text{many}) \leftarrow \\
\mathrm{T}(\epsilon) \leftarrow & \mathrm{D}(\epsilon) \leftarrow \\
\mathrm{VP}(st) \leftarrow \mathrm{V}(s)\ \mathrm{DP}(t) & \mathrm{NP}(s) \leftarrow \mathrm{N}(s) \\
\mathrm{VP}(st) \leftarrow \mathrm{V}(s)\ \mathrm{CP}(t) & \mathrm{N}(\text{students}) \leftarrow \\
\mathrm{V}(\text{know}) \leftarrow & \mathrm{N}(\text{teachers}) \leftarrow \\
\mathrm{CP}(st) \leftarrow \mathrm{C}(s)\ \mathrm{TP}(t) & \\
\mathrm{C}(\text{that}) \leftarrow & \\
\mathrm{C}(\epsilon) \leftarrow &
\end{array}
$$

Standard CFG notation would present that first rule as TP → DP T′, but we use the equivalent logical formulation: If string s is a DP and string t is a T′, then string st is a TP. Similarly, instead of D → some, we write D(some) ←, meaning: *some* is a D. Derivations of sentences can be displayed in a tree such as the following:



Let CFLs be the whole class of languages that can be defined by grammars of this kind, and it is easy to show that RLs⊂CFLs (Hopcroft & Ullman 1969).

In the language defined by the grammar above, the set of NPs is the substitution class [*students*], but the set of TPs is not a substitution class, as we can see by noting, for example, that *C*(*students know teachers*) contains (many, ϵ) whereas *C*(*many students know teachers*) does not.

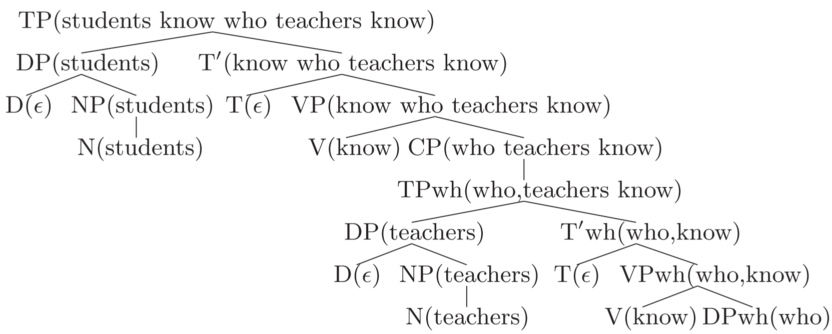
It might seem that things would be simpler if we eliminated the empty T and the empty D, but procedures for eliminating empty categories from CFGs increase grammar size, intuitively "missing generalizations" (Hopcroft & Ullman 1969, theorem 4.11). In linguistic theory, as in any other science, the generalizations are the clues to what is going on, so we do not want to ignore them! Rather, to understand the relation between phrase structure grammars and substitution sets, we need a better analysis, and already we arrive at material that is not yet in any of the introductory texts.

Clark (2015) shows how to connect substitution classes with phrase structure grammars: Instead of considering substitution classes alone, we relativize them to classes of contexts. So, in our example, paired with the set of Ds $\{\epsilon$, some, many$\}$, there is an infinite set of contexts in which those can appear $\{(\epsilon$, students know teachers), (students know, teachers), $\ldots\}$. The (substitution class, context class) pairs we are interested in are maximal in a certain sense, related by a Galois connection. Clark calls these pairs "syntactic concepts" and shows that they form a lattice with respect to the subset relation. For each category A, a grammar defines an associated syntactic concept, a (substitution class, context class) pair. If we let $[A]$ be the substitution class for category A, then for any rule $A \leftarrow BC$, it is easy to show that $[B][C] \subseteq [A]$. Therefore, as expected, what a phrase structure derivation tree shows is how a particular string participates in these these inclusion relationships. Linguists know that each derivation is calculated simply by using the rules, but the derivation matters because of the inclusion relations it reveals. This is one origin of tree structures in syntax, so trees have an origin in string sets after all. Furthermore, ambiguity is expected because an expression can be in multiple substitution classes, each associated with a different context class. And, when context classes are specified, empty strings make sense. This explicit connection between strings and contexts set the stage for the discovery of new learning algorithms (reviewed in Clark 2017).

2.1.1. Discontinuities. Human languages often allow phrases that play a role in one position to play that same role in another, quite different position. Pollard (1984) noticed that this kind of "displacement" or "movement" can easily be added to phrase structure grammars by letting the categories classify not only strings but also pairs (or in general, n-tuples) of strings. Adding the following rules to the previous grammar provides simple *wh*-phrase fronting:

$$\begin{array}{ll} \mathrm{CP}(st) \leftarrow \mathrm{TPwh}(s,t) & \mathrm{VPwh}(u,st) \leftarrow \mathrm{V}(s,t)\ \mathrm{DPwh}(u) \\ \mathrm{TPwh}(t,su) \leftarrow \mathrm{DP}(s)\ \mathrm{T'wh}(t,u) & \mathrm{DPwh}(\mathrm{who}) \leftarrow \\ \mathrm{T'wh}(t,su) \leftarrow \mathrm{T}(s)\ \mathrm{VPwh}(t,u) & \end{array}$$

These rules move *who* to the left edge of CP, as shown below:



If each string variable that appears on the right side of a rule appears at most once on the left, as in this example, then these are multiple context-free grammars (MCFGs), and the string languages they define are MCFLs (Seki et al. 1991). Clark & Yoshinaka (2014) show that the substitution class idea extends naturally to MCFLs. The category T′wh represents a pair of expressions and contexts, both infinite sets, in which each expression now has two parts (s,t), and so each of the contexts has three parts (l,m,r), left–middle–right, such that substituting the parts of the expression into the respective positions in the context is a grammatical sentence.

A thoughtful reader might notice that, intuitively, we should not have needed to add new rules for VP, T′, and TPwh, since VP, T′, and TP have essentially the same distribution whether or not *wh*-phrases are extracted from them. The minimalist grammars inspired by Chomsky (1995) aim to avoid that problem in the simplest way possible (Stabler 1997). That system and its many minor variants can formalize a wide range of proposals in Chomskian syntax (Stabler 2010). Minimalist grammars are expressively equivalent to MCFGs (Harkema 2001, Michaelis 2001) but can be much more succinct (Stabler 2013). It is illuminating to compare the slightly different perspectives on discontinuous constituents provided by recent type-logical (Morrill & Valentín 2010, Wijnholds 2014), and combinatory categorial grammars (Kuhlmann et al. 2015, 2017). Another approach to discontinuity uses "path equations" to constrain related positions in syntactic structures, but in most formulations, these result in systems much more powerful than necessary for human syntax (Johnson 1988, Torenvliet & Trautwein 1995).

2.1.2. Well-nestedness. In MCFGs, the string variables that appear on the right side of a rule can appear in any order on the left side. Intuitively, this allows unbounded scrambling. One artificial language with unbounded scrambling is called MIX. It is a language with only three words, containing all and only the strings in which each of the three words occurs the same number of times. So if the three words are a, b, and c, then $baabcc \in$MIX but $aaabcc \notin$MIX. MIX was recently shown to be an MCFL with categories that classify one or two strings (Salvati 2015). Joshi (1985) conjectures that human languages are not CFLs but rather are "mildly context sensitive" in a sense that disallows MIX.

We can restrict scrambling by imposing restrictions on the positions of variables in an MCFG. In particular, let us say that an MCFG is well-nested (i.e., a WNMCFG) if and only if

- in each rule, every string variable appears exactly once on each side,
- the variables of each term on the right appear in the same order on the left, and
- in any rule A ← B C, if two variables b_1 and b_2 from B are separated on the left side by any variable c from C, then all the variables from C appear between b_1 and b_2, and similarly, if two variables c_1 and c_2 from C are separated on the left side by any variable b from C, then all the variables from B appear between c_1 and c_2.

Intuitively, this means that for any rule A ← B C, we can draw lines over A connecting the B variables, and lines connecting the C variables, without crossing lines. For example, all the rules in our example grammar above are obviously well-nested—the only slightly interesting nesting is in the rules for TPwh and T′wh, where s is properly nested between t and u. Similarly, these rules are well-nested (examples from Salvati 2015):

$$\mathrm{I}(j_1k_1, k_2j_2) \leftarrow \mathrm{J}(j_1, j_2)\ \mathrm{K}(k_1, k_2),$$
$$\mathrm{A}(b_1d_1, d_2b_2c_1, c_2c_3b_3) \leftarrow \mathrm{B}(b_1, b_2, b_3)\ \mathrm{C}(c_1, c_2, c_3)\ \mathrm{D}(d_1, d_2),$$

whereas these are not:

$$\mathrm{I}(j_1k_1, j_2k_2) \leftarrow \mathrm{J}(j_1, j_2), \mathrm{K}(k_1, k_2),$$
$$\mathrm{A}(d_1b_1, c_1b_2d_2c_2b_3, c_3) \leftarrow \mathrm{B}(b_1, b_2, b_3)\ \mathrm{C}(b_1, b_2, b_3)\ \mathrm{D}(d_1, d_2).$$

If one draws a line over $\mathrm{I}(j_1k_1, j_2k_2)$ connecting k_1 and k_2, it will cross the line connecting j_1 and j_2. It is known that MIX is not a WNMCFL, so Joshi's claim is sometimes interpreted as meaning that human languages are WNMCFLs.

The definition of WNMCFLs may seem artificial and slightly complicated to linguists, but certain empirically motivated restrictions guarantee well-nestedness. In particular, as noted above, minimalist grammars as originally defined are equivalent to MCFGs, but linguists have proposed

HIERARCHIES OF STRING LANGUAGES

As described by, for example, Jäger & Rogers (2012), the Chomsky hierarchy classifies string languages by the kind of grammar that can generate them, and recent research highlights the linguistic relevance of classes slightly larger than the CFLs and classes slightly smaller than RLs. If we let $\subset$ mean "is a subset and not equal to," we have the following relations among the classes of languages mentioned in Section 2:

$$\text{RLs} \subset \text{CFLs} \subset \text{WNMCFLs} \subset \text{MCFLs} \subset \text{PMCFLs}.$$

Human languages, considered as string sets, are usually thought to be in one of the three larger classes in this hierarchy. But, as explained in the introduction, it is not easy to avoid the conclusion that, in human languages, every sequence of words is grammatical, which would mean that human languages are RLs. Shieber (1985) claims that the string set of Swiss German is not in CFLs, but he pays no attention to how this claim is threatened by direct quotation. The claim of real interest, one not threatened by direct quotation, is not that human languages are not RLs, but the more abstract claim that the mechanisms needed to define human language structures can define string languages that are not RLs but cannot define string languages that are not PMCFLs. Section 5.1, below, presents another perspective on these classes.

a number of different kinds of restrictions on movements out of specifier positions—"freezing" or "specifier island" or "left branch" constraints (Sauerland 1999, Koopman & Szabolcsi 2000, Collins 2005, Abels 2007). When the specifier island constraint is imposed on minimalist grammars, the definable languages are all WNMCFLs (Kanazawa et al. 2011) (see the sidebar titled Hierarchies of String Languages).

2.1.3. Copying. As noted above, in MCFGs, each variable that appears on the right side of a rule appears exactly once on the left. If we relax that restriction to allow a variable to occur more than once on the left, the effect is that multiple copies of that string value will appear in the result. The extension of MCFGs to allow this is called parallel MCFG (PMCFG), and these grammars are strictly more expressive. Many linguists assume that syntax has copying operations (Koopman 1984, Kobele 2006, Hein 2018), but there are some surprisingly persuasive arguments to the effect that the causes of copying in language are, at least in some cases, not really syntactic (Pullum & Rawlins 2007).

2.2. Substitution Classes in Human Languages

Our most basic substitution-based empirical claim about human languages is as follows:

> **Claim 1**. The substitution class of each sentence can be derived from the substitution classes of its parts, providing a tree-structured analysis for every sentence based on only finitely many substitution classes altogether. Phrase structure grammars (in the generalized sense that includes PMCFGs) define these classes, allowing ambiguity, empty categories, and discontinuity.

If we look at what is needed for the grammars for human languages, it seems that we need only a very restricted kind of grammar. As noted above, Joshi (1985) makes a much stronger claim, but according to the very conservative stance adopted here, the following claim seems likely:

> **Claim 2**. The mechanisms needed to define human languages can define only PMCFLs.

2.3. The Insufficiency of Substitution Classes

Every syntactic category corresponds to a (substitution class, context class) pair, but it is not the case that every such pair corresponds to something we want as a syntactic category. Every substring is in some substitution–context pair, and we obviously do not need all of them in our grammars. In the previous example, we do not need a category for nonconstituents like [*students know*]. So being a substitution–context pair does not suffice for being needed in the syntax. The exactly analogous point applied to earlier notions of substitution classes, as noted by Chomsky (1953, pp. 243–44; 1959, p. 209). What we would like to do is to clearly identify the crucial additional properties of syntactic structures.

2.3.1. Substitution classes with with compositional semantics. We understand new expressions on the basis of knowing their parts and assembly, where those parts and assembly are syntactic [Frege 1977 (1918)]. Furthermore, both language-learning children and syntacticians give a great deal of attention to the meanings of the expressions they are studying, so perhaps what distinguishes the appropriate syntactic categories from the many other possible substitution classes is the availability of an appropriate, compositional interpretation of the derived strings. But the requirement of compositionality by itself, as usually formulated, is not enough. Any language with any semantics at all can be given a compositional semantics that, in some sense, agrees on the semantic values of the sentences (Zadrozny 1994, Westerståhl 2004). Linguists have various intuitions about what kinds of compositional semantics should be regarded as plausible and compositional, and there have been attempts to rigorously formalize some of them (Kracht 2007a, Keenan & Stabler 2010), but the matter remains unclear. Consequently, when semantic values of both wholes and parts are not simply given in advance, it is difficult to use compositionality to rule out an unwanted syntactic analysis, even in clear cases (see Kracht 2007b for one of the rare attempts to do so). With neural net technologies, it is becoming common to use distributional semantics to guide structure identification. There are some intriguing first successes (Williams et al. 2018), but these systems are still rather poorly understood. Still, even substitution classes admitting a compositional semantics are unlikely to define the syntactic constituents of human languages, as the rest of this review argues.

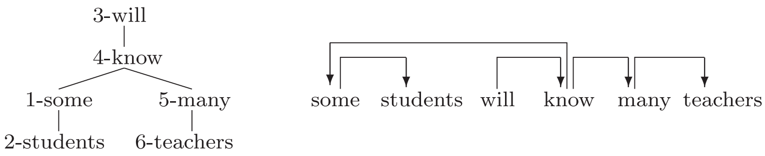
2.3.2. Substitution classes maximizing transition probabilities. Another old idea about substitution classes (e.g., Harris 1951) is roughly that word-to-word transition probabilities should be higher inside constituents than between constituents. Does this explain why [*students know*] is not a constituent but [*know teachers*] is? That idea does not work very well, and many other statistical criteria have been attempted. Klein & Manning (2002) use an expectation maximization algorithm to maximize a carefully designed probability function on trees. Assessing their induced structures for English part-of-speech tag sequences, Klein & Manning (2002) found that their algorithm differs from linguists in systematic ways: It forms a constituent of modal and main verb, it attaches postverbal prepositions to the verb, and so forth. Studies by Klein & Manning (2005), Spitkovsky et al. (2013), and Pate & Johnson (2016), and most other recent research in this tradition, aim to infer grammars that draw many fewer distinctions than linguists typically do, and they also use dependency relations, introduced in the next section.

3. TREES FROM DEPENDENCIES

There is a different idea about human language, completely missed in the whole previous section. That section chunks strings of words into hierarchies of substitution classes, but substitution classes, categorized by typically rather complex phrase structure grammars, may be providing the wrong perspective on what is really happening in human language. Instead, maybe words enter directly into relations with each other. Many but not all linguists in this tradition assume that the dependencies form a tree.

3.1. Dependency Trees

Numbering the positions of the words of any string from 1 to n, let us say that a dependency tree is a tree over the numbered word positions (Kuhlmann & Satta 2009). The dependency tree shown on the left below, for example, is often displayed as shown on the right, with word positions indicated by left-to-right order:



Each node except one (the root) has a unique incoming arc from its head. The children of any node, indicated by outgoing arcs, are dependents. To emphasize similarity with the earlier phrase structure grammar, this example structure makes *some* the head of *students*, but the opposite is usually assumed in dependency grammar. And while dependency grammars assume each head is a pronounced lexical item, many linguists proposing phrase structure grammars allow empty heads too. Finally, often the arcs of the tree are classified and labeled accordingly; for example, the arc from *know* to *some* could be labeled subject or specifier, the arc from *know* to *many* could be labeled object or complement, and so on. A dependency language is any set of dependency trees (possibly infinite). The parsing problem, then, is to compute, for any string, all and only the dependency structures that the string has in the dependency language. And the language acquisition problem for dependency languages is to acquire some specification of this set (implicitly, some kind of grammar). In many recent engineering approaches, head–dependent relations are learned from example dependency structures and generalized to new instances on the basis of distributional properties, with a check that the relations form a tree that spans the sentence (McDonald et al. 2005, Kiperwasser & Goldberg 2016, Dozat & Manning 2017).

3.2. Dependencies in Human Languages

We can formulate a conservative claim about dependencies as follows:

> **Claim 3**. In a well-formed sentence, each word other than the unique root is a "dependent" of some "head." When each word other than the root has at most one head, this again provides a tree-structured analysis.

Obviously the basic Claim 1 does not entail Claim 3 or anything about dependencies at all. And although Claim 3 entails that language has a tree structure, it does not tell us that the constituents in that tree structure (each head with its dependents) correspond to substitution classes, let alone that the language could be defined with only finitely many of those substitution classes. So Claims 1 and 3 are truly logically independent, and compatible.

3.3. The Insufficiency of Dependencies, and Mixed Theories

Notice that Claim 3 tells us that human languages have heads and dependents, but we have not imposed any precise requirements on what it is to be a head or dependent. Lacking restrictions on what should be a head or dependent, a description of a language in these terms obviously does not suffice for being relevant to the syntax of human languages.

The definition of a dependency language given above allows languages that have no finite specification, but we can impose restrictions on dependency languages by requiring that the language be closed with respect to certain dependency assignment rules. Kuhlmann & Satta (2009) and Kuhlmann (2013) provide one way of doing so. Their method guarantees that the dependencies can be defined with finitely many rules, and allows precise connections to phrase structure grammars to be established.

In fact, most substitution-based grammars also identify a head in each substitution-class-based constituent, an element whose role is in some sense "most important" in determining the internal structure of the constituent (its dependents) and the external distribution of the constituent in the rest of the language. So these are "mixed theories" in the sense that they have both Claims 1 and 3 in their foundations. In the example grammar shown above, the rules for D and DP are similar to those for V and VP and those for N and NP, and so forth. Therefore, instead of phrase structure rules for each instance, we could use a general schema and attribute the necessary variations on that scheme to properties of the heads D, V, N, and so on. X-bar Theory is one version of this kind of proposal (Chomsky 1970). A purer version of dependency theory emerges in more recent proposals about "bare syntax" (Chomsky 1995, p. 226, ex. 8*b*). The categories of type-logical and categorial grammars (Moortgat 2010, Steedman 2014) and Minimalist grammar (Stabler 1997) transparently display the definition of internal structure and external distribution by heads, and their derivations can be displayed explicitly as dependency trees (Boston et al. 2010). In these mixed theories, each lexical item determines part of a derivation tree, so it is no surprise that explicitly tree-based tree-adjoining grammars have similar properties (Joshi & Schabes 1997, Kallmeyer 2010).

4. TREES FROM OPTIMIZING PREFERENCES

The previous sections seem to miss almost all of the most obvious properties of human languages that anyone with common sense would notice. First, the previous sections tell us nothing about sizes: For example, speaking very roughly, the vocabulary of a human speaker is less than a million words; no language has only or even mainly (phonetically, gesturally) empty words; the number of substitution classes needed to specify a human grammar is less than a billion; the number of obligatory dependents of a head seems never to exceed four; the number of overt distinct case markings and distinct genders and distinct tenses is always less than 100. Second, there are systematic frequency effects that the previous sections would not predict: In the sentences of any normal discourse, the average number of words per clause is less than 50; the average depth of center-embedded clauses is less than three; the average number of fronted *wh*-phrases per clause is less than three; and in any normal discourse, the relation between rank and frequency for words, for substitution classes, and for head–dependent pairs approximates an inverse exponential distribution (Zipf 1949). Third, there are numerous robustly evidenced structural tendencies that would be completely unexpected given only the perspectives of the previous sections: Discontinuous constituents rarely include one element inside a *wh*-clause and one outside of that clause (Ross 1967, Rizzi 1978, Chomsky 1986, Cinque 1990, Pesetsky 2000, Starke 2001, Boeckx 2012); focused constituents tend to be at the left edge of the clause (Rizzi 1997, Pan 2015); some orders of [Dem Num Adj N] are much more common than others (Cinque 2005); affix/constituent order mirrors hierarchy/semantic relatedness/grammaticality (Baker 1985, Kayne 1994, Brody 1998, Koopman 2005, Myler 2017); and the order of movement types mirrors phrase structure (Hiraiwa 2005, Abels & Neeleman 2006, Boeckx 2008). Furthermore, when examining preferences at a finer level of detail, we find cases in which languages that block a preferred property for some reason sometimes nevertheless conform to the preference in exceptional cases [i.e., "emergence of the unmarked" (Bresnan 1997, Costa 2001, Müller 2001, Vogel 2004)]. Clearly, a wide range of diverse factors is needed to explain these things, but some of these tendencies may be due to fundamental, universal preferences, and it could be appropriate to formulate syntactic theory in a way that recognizes those preferences explicitly.

4.1. Optimization-Based Grammars

There are many variants of Optimality-Theoretic (OT) grammars, inspired by Prince & Smolensky (2004) and related research. Adapting a formalization from Frank & Satta (1998), define OT grammar G as follows. Assume we have a function Gen that maps strings to a set of candidate syntactic structures. And assume we have an ordered sequence of $n > 1$ constraints $\mathbf{c} = c_1, \ldots, c_n$, each of which is a function from structures to integers, intuitively, mapping each structure to the number of times that it violates the constraint. We can then think of the sequence of constraints $\mathbf{c}$ as a function that maps any structure t to a vector of integers $\mathbf{c}(t) = \mathbf{i} = (i_1, \ldots, i_n)$, where each $i_j = c_j(t)$. For any two vectors of integers $\mathbf{i}$ and $\mathbf{j}$, let us say $\mathbf{i} < \mathbf{j}$ if and only if for some $0 < k \leq n$, $i_k < j_k$ and for all $\ell < k$, $i_\ell \leq j_\ell$. Then, G maps each string s to the subset of $Gen(s)$ for which $\mathbf{c}(s)$ is minimal:

$$G(s) = \{t \in Gen(s) | \mathbf{c}(t) = \min\{\mathbf{c}(\mathrm{T}') | \mathrm{T}' \in Gen(s)\}\}.$$

This and similar frameworks find a very wide range of applications in linguistic theory. Of particular interest is the way violations of preferences can indicate to a learner which preferences take priority (Tesar & Smolensky 2000, Enguehard et al. 2018), and how optimization could be psychologically, neurally realized (Smolensky 1986, Smolensky et al. 2014, Cho et al. 2018).

4.2. Optimization in Human Languages

We can make a very conservative claim about preference optimization in language as follows:

> **Claim 4**. Syntactic structures optimize preferences. In effect, syntactic preferences define a sense of "simplest" ("optimal," "most harmonic," "most economical") in which the simplest analyses are preferred.

The empirical claim here is, in effect, simply that OT or some similar framework that incorporates preference optimization will have application in syntax. We can also note that when the simplest analyses of very large or infinite domains are compositional, this may again provide a different kind of motivation for tree-structured analyses.

It is obvious that Claim 4 says nothing about substitution classes or dependencies. And the earlier Claims 1 and 3 say nothing about preferences. So these three ideas are all independent of one another.

4.3. The Insufficiency of Preference Optimization, and Mixed Theories

Smolensky (1986, p. 196) makes the point that an optimality framework such as the one described in Section 4.1 is a mathematical theory, not a scientific one. Claim 4 in Section 4.2 makes the very weak empirical claim that the framework is appropriate for at least some syntactic phenomena. The real empirical content comes from the specification of preferences, constraints that we would prefer not to violate, just as Claim 2 gets its substance from the specification of particular dependencies and Claim 1 gets its substance from the specification of particular phrase structure rules.

Preferences are obviously compatible with the previous ideas about language. Although the relation between constraints and properties of rule-based languages can sometimes be nonobvious, it is no surprise that some kinds of constraints define analyses equivalent to certain kinds of substitution-based grammars (Frank & Satta 1998, Hao 2017). Graf (2013) shows that some economy constraints in syntax, originally thought to be completely intractable, are enforceable in phrase structure grammars that define efficiently parsable languages. Graf & Heinz (2015) define a class of Minimalist grammars equivalent to MCFGs whose derivation trees form subregular tree languages that are "tier based strictly local," the analog of a phonologically motivated subregular class of strings.

If we consider which of our three claims is most fundamental, it is certainly this one, and it seems likely a slight enrichment of Claim 4 could subsume both Claims 1 and 3. But there are many issues to settle before we can see exactly how that should be done.

5. THE COINCIDENCE

5.1. The Importance of Tree Languages

A set of trees is sometimes called a forest, but we stick to the term tree language. Given the importance of tree languages in linguistic theory, they deserve more attention than they receive. They reveal a simple progression from simple regular grammars to CFGs to PMCFGs, and PMCFGs can (probably) define all linguistic structures (see the sidebar titled The Myhill–Nerode Theorem for Tree Languages). And as with string languages, hierarchies of tree languages can be distinguished by the kinds of definitions required, related to the kinds of devices that can recognize them (see the sidebar titled Hierarchies of Tree Languages).

5.2. Explaining the Coincidence

Rather than presenting three different ideas, why not present a single framework that perfectly integrates them? First, it is important to understand that there are different ideas, and the suggestion here is that they are correct but just not <u>complete</u>, not <u>sufficient</u>. Much is poorly understood in linguistics, but that does not mean that these ideas will have to be thrown out.

If these three ideas are independent and correct, then their coincidence needs to be explained. Section 4.3 speculates that a slight reformulation of these principles might allow one of them to subsume the others. It is also plausible that one basic and simple ability, one simple computational system, could properly realize them. This is already suggested by the brief remarks about "mixed views" above.

There is greater consensus about the integration of these ideas than is generally recognized. Many different kinds of grammar define headed, substitution-class-based, semantically

THE MYHILL–NERODE THEOREM FOR TREE LANGUAGES

Kozen (1992) presents, in a very simple, accessible form, the natural extension of the Myhill–Nerode theorem to tree languages. In any tree, we can replace any subtree by a variable to obtain a context of that subtree. Then, the theorem says that a tree language is regular (i.e., accepted by a finite-state bottom-up automaton) if and only if, in that language, the number of context sets $C(t)$ of subtrees t is finite.

HIERARCHIES OF TREE LANGUAGES

Let RTs be the class of tree languages recognized by finite-state bottom-up acceptors. Let 1-RTs be the restriction of RTs to nonbranching trees, and let MRTs be the class of tree languages recognized by finite-state bottom-up acceptors whose states classify not just subtrees but tuples of subtrees (the *M* stands for multi-). Then, obviously, we have

$$\text{1-RTs} \subset \text{RTs} \subset \text{MRTs}.$$

Since a nonbranching tree is a string, 1-RTs represent simply another perspective on RLs. And for each language in RTs, if we consider the corresponding string languages formed by the leaves of its trees, these corresponding string languages are exactly the CFLs (Thatcher 1967). For MRTs, if we consider the corresponding string languages, they are the MCFLs (Seki et al. 1991, Engelfriet et al. 2009). Adding output to tree acceptors yields tree transducers. For any Minimalist grammar, the set of its derivation trees is in RTs, and a finite-state multi-bottom-up tree trandsucer maps each derivation tree to a tree in which, intuitively, the movements have been performed (Kobele et al. 2007). Some other mildly context-sensitive grammar formalisms allow similar two-step formulations (Mönnich 1998, Morawietz 2001). There is a rich theory here, with interesting and sometimes surprising hierarchies of complexity relations.

compositional, tree-like structures that can be shaped by preferences. One advantage of mathematical perspectives is that abstract and sometimes nonobvious similarities between different formal systems can be recognized and precisely characterized. Another advantage of precise mathematical formulations is that they can illuminate places where these systems most directly engage empirically testable claims.

One area in which mathematically precise formulations are essential and are already leading to important insights is in models of language acquisition, since those must define how grammars are shaped by the impact of experience. As we discover how to connect more linguistic phenomena to precisely formulated foundations, we will understand more about how language acquisition can happen. As every linguist knows, there are very many systematic linguistic phenomena that we do not know how to properly connect.

SUMMARY POINTS

1. Mainstream syntax mainly embraces these claims:
   - The substitution class of each phrase can be derived from the substitution classes of its parts, providing a tree-structured analysis for every sentence. Phrase structure grammars (in the generalized sense that includes PMCFGs) define these classes, allowing ambiguity, empty categories, and discontinuity.
   - If each word in a sentence has at most one head, and each sentence has at most one "root" element with no head, then this again provides a tree-structured analysis.
   - Linguistic structures optimize linguistic preferences. In effect, the preferences define a sense of "simplest" in which simplest analyses are preferred. When the simplest analyses of very large or infinite domains are compositional, this may again motivate tree-structured analyses.

2. The mechanisms needed to define human languages can define only PMCFLs and perhaps they are even "mildly context sensitive" in the sense that they can define only WNMCFLs.
3. In some mildly context-sensitive formalizations of recent syntax, the set of derivations is a regular tree language, and the effect of movement is a regular transduction. This reveals that human languages can be defined in two finite-state steps.

FUTURE ISSUES

1. With syntax given as a finite-state transducer, we may be able to achieve a simple and illuminating integration of the interfaces, as imagined by Kaplan & Kay (1994, pp. 337, 377): "Repeated composition reduces the machines corresponding to the rules of a complete phonological grammar to a single transducer.... It might therefore prove advantageous to seek ways of composing phonology and syntax to produce a new system with the same formal properties as syntax alone."
2. This review presents three independent claims about language. Can a slight enrichment of our weak optimization claim subsume the claims about substitution classes and dependencies to provide a unified foundation for linguistic theory? The conjecture offered here is: yes.

DISCLOSURE STATEMENT

The author is not aware of any funding or financial holdings that might be perceived as affecting the objectivity of this review.

LITERATURE CITED

Abels K. 2007. Towards a restrictive theory of (remnant) movement: improper movement, remnant movement, and a linear asymmetry. *Linguist. Var. Yearb.* 7:53–120
Abels K, Neeleman A. 2006. *Universal 20 without the LCA*. Work. pap., Univ. Coll. London, UK
Adger D. 2003. *Core Syntax: A Minimalist Approach*. Oxford, UK: Oxford Univ. Press
Baker MC. 1985. The mirror principle and morphosyntactic explanation. *Linguist. Inq.* 16:373–416
Boeckx C. 2008. *Bare Syntax*. New York: Oxford Univ. Press
Boeckx C. 2012. *Syntactic Islands*. New York: Cambridge Univ. Press
Boston MF, Hale JT, Kuhlmann M. 2010. Dependency structures derived from Minimalist grammars. In *Mathematics of Language 10/11*, ed. C Ebert, G Jäger, J Michaelis, pp. 1–12. Lect. Notes Comput. Sci. vol. 6149. Berlin: Springer
Bresnan J. 1997. The emergence of the unmarked pronoun: Chichewâ pronominals in Optimality Theory. In *Proceedings of the 23rd Annual Meeting of the Berkeley Linguistics Society: Special Session on Syntax and Semantics in Africa*, pp. 26–46. Berkeley, CA: Berkeley Linguist. Soc.
Bresnan J, Asudeh A, Toivonen I, Wechsler S. 2015. *Lexical–Functional Syntax*. Oxford, UK: Wiley-Blackwell. 2nd ed.
Brody M. 1998. Projection and phrase structure. *Linguist. Inq.* 29:367–98
Cho PW, Goldrick M, Lewis RL, Smolensky P. 2018. Dynamic encoding of structural uncertainty in gradient symbols. In *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics* (*CMCL*), pp. 19–28. Stroudsburg, PA: Assoc. Comput. Linguist.
Chomsky N. 1953. Systems of syntactic analysis. *J. Symb. Log.* 18:242–56

Chomsky N. 1959. Review of Joseph H. Greenberg, *Essays in Linguistics*. *Word* 15:202–18

Chomsky N. 1970. Remarks on nominalization. In *Readings in English Transformational Grammar*, ed. RA Jacobs, PS Rosenbaum, pp. 184–221. Boston: Ginn

Chomsky N. 1986. *Barriers*. Cambridge, MA: MIT Press

Chomsky N. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press

Chomsky N. 2006. *Language and Mind*. New York: Cambridge Univ. Press. 3rd ed.

Cinque G. 1990. *Types of A′ Dependencies*. Cambridge, MA: MIT Press

Cinque G. 2005. Deriving Greenberg's Universal 20 and its exceptions. *Linguist. Inq.* 36:315–32

Clark A. 2015. The syntactic concept lattice: another algebraic theory of the context-free languages? *J. Log. Comput.* 25:1203–29

Clark A. 2017. Computational learning of syntax. *Annu. Rev. Linguist.* 3:107–23

Clark A, Yoshinaka R. 2014. An algebraic approach to multiple context-free grammars. In *Proceedings of the International Conference on Logical Aspects of Computational Linguistics*, pp. 57–69. Berlin: Springer

Collins C. 2005. A smuggling approach to passive in English. *Syntax* 8:81–120

Costa J. 2001. The emergence of unmarked word order. In *Optimality-Theoretic Syntax*, ed. G Legendre, JB Grimshaw, S Vikner, pp. 171–204. Cambridge, MA: MIT Press

Dozat T, Manning CD. 2017. *Deep biaffine attention for neural dependency parsing*. Paper presented at the 5th International Conference on Learning Representations, Toulon, Fr., April 24–26

Engelfriet J, Lilin E, Maletti A. 2009. Composition and decomposition of extended multi bottom-up tree transducers. *Acta Inf.* 46:561–90

Enguehard É, Flemming E, Magri G. 2018. Statistical learning theory and linguistic typology: a learnability perspective on OT's strict domination. In *Proceedings of the Society for Computation in Linguistics* (*SCiL 2018*), pp. 1–11. Stroudsburg, PA: Assoc. Comput. Linguist.

Frank R, Satta G. 1998. Optimality theory and the generative complexity of constraint violability. *Comput. Linguist.* 24:307–15

Frege G. 1977 (1918). Thoughts. In *Gottlob Frege: Logical Investigations*, transl. PT Geach, RH Stoothof. New Haven, CT: Yale Univ. Press

Graf T. 2013. *Local and transderivational constraints in syntax and semantics*. PhD thesis, Univ. Calif., Los Angel.

Graf T, Heinz J. 2015. *Commonality in disparity: the computational view of syntax and phonology*. Slides presented at the 38th Generative Linguistics in the Old Word Colloqium (GLOW 38), Paris, April 15–18

Hao Y. 2017. Harmonic Serialism and finite-state Optimality Theory. In *Proceedings of the 13th International Conference on Finite State Methods and Natural Language Processing*, pp. 20–29. Stroudsburg, PA: Assoc. Comput. Linguist.

Harkema H. 2001. A characterization of minimalist languages. In *Proceedings of the International Conference on Logical Aspects of Computational Linguistics*, pp. 193–211. Berlin: Springer

Harris ZS. 1951. *Methods in Structural Linguistics*. Chicago: Chicago Univ. Press

Hein J. 2018. *Verbal fronting: typology and theory*. PhD thesis, Univ. Leipzig, Leipzig, Ger.

Hiraiwa K. 2005. *Dimensions of symmetry in syntax: agreement and clausal architecture*. PhD thesis, MIT, Cambridge, MA

Hopcroft JE, Ullman JD. 1969. *Formal Languages and Their Relation to Automata*. Reading, MA: Addison-Wesley

Jäger G, Rogers J. 2012. Formal language theory: refining the Chomsky hierarchy. *Philos. Trans. R. Soc. B* 367:1956–70

Johnson M. 1988. *Attribute Value Logic and The Theory of Grammar*. Stanford, CA: Cent. Study Lang. Inf.

Joshi AK. 1985. How much context-sensitivity is necessary for characterizing structural descriptions. In *Natural Language Processing: Theoretical, Computational and Psychological Perspectives*, ed. D Dowty, L Karttunen, A Zwicky, pp. 206–50. New York: Cambridge Univ. Press

Joshi AK, Schabes Y. 1997. Tree-adjoining grammars. In *Handbook of Formal Languages*, vol. 3: *Beyond Words*, ed. G Rozenberg, A Salomaa, pp. 69–124. Berlin: Springer

Kallmeyer L. 2010. *Parsing Beyond Context-Free Grammars*. Berlin: Springer

Kanazawa M, Michaelis J, Salvati S, Yoshinaka R. 2011. Well-nestedness properly subsumes strict derivational minimalism. In *Proceedings of the International Conference on Logical Aspects of Computational Linguistics*, pp. 112–28. Berlin: Springer

Kaplan R, Kay M. 1994. Regular models of phonological rule systems. *Comput. Linguist.* 20:331–78

Kayne RS. 1994. *The Antisymmetry of Syntax*. Cambridge, MA: MIT Press

Keenan EL, Stabler EP. 2010. Language variation and linguistic invariants. *Lingua* 120:2680–85

Kiperwasser E, Goldberg Y. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Trans. Assoc. Comput. Linguist.* 4:313–27

Klein D, Manning CD. 2002. A generative constituent-context model for improved grammar induction. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 128–35. Stroudsburg, PA: Assoc. Comput. Linguist.

Klein D, Manning CD. 2005. Corpus-based induction of syntactic structure: models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pp. 478–85. Stroudsburg, PA: Assoc. Comput. Linguist.

Kobele GM. 2006. *Generating copies: an investigation into structural identity in language and grammar*. PhD thesis, Univ. Calif., Los Angel.

Kobele GM, Retoré C, Salvati S. 2007. An automata-theoretic approach to minimalism. In *Proceedings of the 2007 European Summer School on Logic, Language and Information (ESSLLI 2007): Model Theoretic Syntax at 10*, pp. 71–80. Stroudsburg, PA: Assoc. Comput. Linguist.

Koopman H. 1984. *The Syntax of Verbs: From Verb Movement Rules in the Kru Languages to Universal Grammar*. Dordrecht, Neth.: Foris

Koopman H. 2005. On the parallelism of DPs and clauses in Kisongo Maasai. In *Verb First*, ed. A Carnie, S Dooley, H Harley, pp. 281–301. Philadelphia: Benjamins

Koopman H, Szabolcsi A. 2000. *Verbal Complexes*. Cambridge, MA: MIT Press

Kozen D. 1992. On the Myhill–Nerode theorem for trees. *Bull. Eur. Assoc. Theor. Comput. Sci.* 47:254–66

Kracht M. 2007a. Compositionality: the very idea. *Res. Lang. Comput.* 5:287–308

Kracht M. 2007b. The emergence of syntactic structure. *Linguist. Philos.* 30:47–95

Kuhlmann M. 2013. Mildly non-projective dependency grammar. *Comput. Linguist.* 39:355–87

Kuhlmann M, Koller A, Satta G. 2015. Lexicalization and generative power in CCG. *Comput. Linguist.* 41:215–47

Kuhlmann M, Satta G. 2009. Treebank grammar techniques for non-projective dependency parsing. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 478–86. Stroudsburg, PA: Assoc. Comput. Linguist.

Kuhlmann M, Satta G, Jonsson P. 2017. On the complexity of CCG parsing. arXiv:1702.06594 [cs]

Manaster-Ramer A. 1983. The soft formal underbelly of theoretical syntax. In *Papers from the 19th Regional Meeting of the Chicago Linguistics Society*, pp. 256–62. Chicago: Chicago Linguist. Soc.

McDonald R, Pereira F, Ribarov K, Hajič J. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the 2005 Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT 05)*, pp. 523–30. Stroudsburg, PA: Assoc. Comput. Linguist.

Merchant J, Frazier L, Clifton C, Weskott T. 2013. Fragment answers to questions: a case of inaudible syntax. In *Brevity*, ed. L Goldstein, pp. 21–35. New York: Oxford Univ. Press

Michaelis J. 2001. Transforming linear context free rewriting systems into Minimalist grammars. In *Proceedings of the International Conference on Logical Aspects of Computational Linguistics*, pp. 228–44. Berlin: Springer

Mönnich U. 1998. TAGs M-constructed. In *Proceedings of the 4th International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+4)*, pp. 108–11. Philadelphia: Inst. Res. Cogn. Sci.

Moortgat M. 2010. Typelogical grammar. In *Stanford Encyclopedia of Philosophy*, ed. EN Zalta. Stanford, CA: Stanford Univ. Winter ed. **https://plato.stanford.edu/entries/typelogical-grammar/**

Morawietz F. 2001. *Two-step approaches to natural language formalisms*. PhD thesis, Univ. Tübingen, Tübingen, Ger.

Morrill G, Valentín O. 2010. Displacement calculus. *Linguist. Anal.* 36:167–92

Müller G. 2001. Order preservation, parallel movement, and the emergence of the unmarked. In *Optimality-Theoretic Syntax*, ed. G Legendre, JB Grimshaw, S Vikner, pp. 279–313. Cambridge, MA: MIT Press

Myler N. 2017. Exceptions to the mirror principle and morphophonological action at a distance. In *The Structure of Words at the Interfaces*, ed. H Newell, M Noonan, G Piggott, L Travis, pp. 100–25. Oxford, UK: Oxford Univ. Press

Pan VJ. 2015. Mandarin peripheral construals at the syntax–discourse interface. *Linguist. Rev.* 32:819–68

Pate JK, Johnson M. 2016. Grammar induction from (lots of) words alone. In *Proceedings of the 26th International Conference on Computational Linguistics* (*COLING2016*), pp. 23–32. Stroudsburg, PA: Assoc. Comput. Linguist.

Pesetsky D. 2000. *Phrasal Movement and Its Kin*. Cambridge, MA: MIT Press

Pollard C. 1984. *Generalized phrase structure grammars, head grammars and natural language*. PhD thesis, Stanford Univ., Stanford, CA

Prince A, Smolensky P. 2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Malden, MA: Blackwell

Pullum GK, Rawlins K. 2007. Argument or no argument? *Linguist. Philos.* 30:277–87

Radford A. 2009. *An Introduction to English Sentence Structure*. Cambridge, UK: Cambridge Univ. Press

Rizzi L. 1978. Violations of the *wh*-island constraint in Italian and the subjacency condition. *Montreal Work. Pap. Linguist.* 11:155–90

Rizzi L. 1997. The fine structure of the left periphery. In *Elements of Grammar*, ed. L Haegeman, pp. 281–337. Boston: Kluwer

Ross JR. 1967. *Constraints on variables in syntax*. PhD thesis, MIT, Cambridge, MA

Sag I, Wasow T, Bender E. 2003. *Syntactic Theory: A Formal Introduction*. Stanford, CA: Cent. Study Lang. Inf. 2nd ed.

Sakarovitch J. 2009. *Elements of Automata Theory*. New York: Cambridge Univ. Press

Salomaa A. 1973. *Formal Languages*. New York: Academic

Salvati S. 2015. Mix is a 2-MCFL and the word problem in Z2 is captured by the IO and the OI hierarchies. *J. Comput. Syst. Sci.* 81:1252–77

Sauerland U. 1999. Erasability and scrambling. *Syntax* 2:161–88

Seki H, Matsumura T, Fujii M, Kasami T. 1991. On multiple context-free grammars. *Theor. Comput. Sci.* 88:191–229

Shieber SM. 1985. Evidence against the context-freeness of natural language. *Linguist. Philos.* 8:333–44

Smolensky P. 1986. Information processing in dynamical systems: foundations of harmony theory. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1: *Foundations*, ed. DE Rumelhart, JL McClelland, pp. 194–281. Cambridge, MA: MIT Press

Smolensky P, Goldrick M, Mathis D. 2014. Optimization and quantization in gradient symbol systems: a framework for integrating the continuous and the discrete in cognition. *Cogn. Sci.* 38:1102–38

Spitkovsky VI, Alshawi H, Jurafsky D. 2013. *Grammar induction and parsing with dependency-and-boundary models*. PhD thesis, Stanford Univ., Stanford, CA

Sportiche D, Koopman H, Stabler E. 2014. *An Introduction to Syntactic Analysis and Theory*. Oxford, UK: Blackwell

Stabler EP. 1997. Derivational minimalism. In *Proceedings of the International Conference on Logical Aspects of Computational Linguistics*, pp. 68–95. Berlin: Springer

Stabler EP. 2010. Computational perspectives on minimalism. In *Oxford Handbook of Minimalism*, ed. C Boeckx, pp. 617–41. Oxford, UK: Oxford Univ. Press

Stabler EP. 2013. Two models of minimalist, incremental syntactic analysis. *Top. Cogn. Sci.* 5:611–33

Starke M. 2001. *Move dissolves into merge: a theory of locality*. PhD thesis, Univ. Geneva, Geneva, Switz.

Steedman MJ. 2014. Categorial grammar. In *Routledge Handbook of Syntax*, ed. A Carnie, Y Sato, D Siddiqi, pp. 670–701. New York: Routledge

Tesar B, Smolensky P. 2000. *Learnability in Optimality Theory*. Cambridge, MA: MIT Press

Thatcher JW. 1967. Characterizing derivation trees of context-free grammars through a generalization of finite automata theory. *J. Comput. Syst. Sci.* 1:317–22

Torenvliet L, Trautwein M. 1995. A note on the complexity of restricted attribute-value grammars. In *Proceedings of Computational Linguistics in the Netherlands* (*CLIN5*), pp. 145–64. Twente, Neth.: Dep. Comput Sci., Univ. Twente

Vogel R. 2004. Correspondence in OT syntax and minimal link effects. In *Minimality Effects in Syntax*, ed. A Stepanov, G Fanselow, R Vogel, pp. 401–41. Berlin: Mouton de Gruyter

Weir A. 2014. *Fragments and clausal ellipsis*. PhD thesis, Univ. Mass., Amherst

Westerståhl D. 2004. On the compositional extension problem. *J. Philos. Log.* 33:549–82
Wijnholds GJ. 2014. Conversions between D and MCFG: logical characterizations of the mildly context-sensitive languages. *Comput. Linguist. Neth. J.* 4:137–48
Williams A, Drozdov A, Bowman SR. 2018. Do latent tree learning models identify meaningful structure in sentences? *Trans. Assoc. Comput. Linguist.* 6:253–67
Zadrozny W. 1994. From compositional to systematic semantics. *Linguist. Philos.* 17:329–42
Zipf GK. 1949. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Boston: Houghton, Mifflin

Annual Review of Linguistics

Volume 5, 2019

Contents

Errata

An online log of corrections to *Annual Review of Linguistics* articles may be found at http://www.annualreviews.org/errata/linguistics