

Ambiguous (α)Merge, Symmetry, and Group Theory: A note on the ideal structure of a hypothetical operation

Joshua Bowles

June 27, 2010

Abstract

This paper is a speculative work. It is concerned with formal foundational issues in syntactic theory. The properties and implications of the operation Merge are reviewed in both a linguistic and computational theory setting. A brief sketch is given for the historical background of formalist thinking from which Merge developed, and the operation ambiguous Merge, here called α Merge, is defined in an algebraic group theoretic framework. The idealized and mathematical framework in which α Merge is defined is broadened to include current linguistic interest in Fibonacci patterns as well as non-linguistic interest in the Church-Turing Thesis. Within this context, a detailed comparison is given between the output of general phrase structure relations for α Merge and for results in [Medeiros \(2008\)](#). From this, a translational symmetry between constituents at phrase and head levels is observed and defined as the “X-ratio.” Next follows the argument that given multiple kinds of symmetry observed in syntax—generally of syntactic objects *and* rules—it is reasonable to investigate the mathematical study of symmetry—Group Theory—for use in syntax. This paper also proposes a corollary to a weakened version of the Strong Minimalist Thesis and discusses basic assumptions about the use of formal models and computability in the study of natural language—to which research of symmetry in syntax must be accountable. Finally, given a group theoretical definition of the operation α Merge, some phase-based implications are presented.

Keywords: Symmetry, Group Theory, Merge, Fibonacci pattern

1 Introduction

Natural language cannot be exhaustively modeled by a formal system such as set theory or first order predicate logic. Early work on language stemming from the “linguistic turn” in philosophy of science (e.g., [Ayer \(1936\)](#), [Carnap \(1937\)](#), [Reichenbach \(1947\)](#), [Tarski \(1956\)](#), [Popper \(1959\)](#), [Quine \(1953\)](#)) coupled with the structuralist tradition in linguistics from Bloomfield, Sapir, and Harris, quickly showed that

an efficient and consistent model of grammar could only be gotten in one of two ways: (a) a synthetic grammar contrived from a first order predicate calculus, or (b) accounting for the distribution of particular morphosyntactic elements in produced language data (see Tomalin (2006) for an account of the formal foundations of modern linguistics). Neither (a) nor (b) shed any explanatory light on what was to quickly become the driving insight of the “new school” of linguistics: that any human being of sufficiently young age could acquire any human language with a statistically insignificant measure of data (i.e., the poverty of stimulus).¹

Eventually the conceptual tensions between (a) and (b), as well as the vague notions of *competence* and *performance* underlying the “new school,” forced a distinction between Internal and External structure known as I-language and E-language. The focus of most generative syntacticians is I-language; their goal is to exhaustively describe and explain the internal mechanisms responsible for initial language acquisition for any and all human languages. In the current climate of Minimalist research it is now more appropriate, following ?, to talk of the “growth form” in terms of acquisition. That is, I-language, as a growth form unique to the human organism, is a natural object. As such, it can be modeled by various formal or mathematical systems with varying degrees of success; relative to specific goals. Here I want to emphasize the point first made in Chomsky (1955), contra Bar-Hillel (1954), that natural language is not a *formal* object. Natural language is—at a specifically defined level of analysis—an object of nature that is subject to constraints, laws, and tendencies of the patterns of nature; see also Chomsky (1995b) and Boeckx & Piattelli-Palmarini (2005). Consequently, we should not expect any formal/mathematical system to exhaustively model natural language; but useful methods for mathematicization should always motivate us.²

As it stands now, with recent formulations of X-bar theory and interest in Fibonacci (F) patterns arising from ideal iterations in both prosody and phrase structures, a special role for rigorously modeling the computability requirements of producing iterated patterns emerges.³ That is, F patterns produce self-similar struc-

¹These issues are still at the center of research for theoretical syntax, natural language processing, and artificial intelligence.

²The implication here is very clear, at least to me: linguists will necessarily have to construct the formalisms needed to model natural language capacity. No object or phenomenon in nature is inherently formal (unless one is a strict Platonist), but such natural objects or phenomena are amenable to formal modeling. This modeling may not be precisely exhaustive, it only needs to approximate enough precision to be useful. This is why integral and differential calculus is so important: it allows one to ignore small parts of imprecision that have negligible effects on the outcomes or predictions of one’s model. The same can be said, perhaps with more zeal, for probability and statistics. However, recognition of the reality of imprecise and inexhaustive models does not stop one from *attempting* exhaustive precision models. We should assume the same methodology applies to linguistics in general.

³Not to mention, the majority of linguists, despite theoretical persuasion, would probably agree that natural language exploits iterated patterns in general. Whether these patterns emerge as a consequence of the laws of physics and biology (Minimalist syntax/semantics), or from cultural patterns related to general human cognition (pragmatics) is beside the point. Focusing on iterated

tures along repeated and computable iterations—and, computable iterations of the F sequence can produce idealized X-bar structures, as [Medeiros \(2008\)](#) and [Soschen \(2008\)](#) have shown. These idealized X-bar structures also map projections of phase domains that have implicational patterns, as ? discuss. Because F sequences are computable, iterated, self-similar generations they also produce symmetrical forms. If one takes the F sequence of X-bar structures and compares two different levels, head (X^0) and phrase (XP), a ratio can be derived that shows symmetry between them; I call this the X-ratio in § 2.2. In the context of these developments it is only natural to look towards the formal study of symmetry—Group Theory—for possible rigor. Most exciting, and necessary, is that if one can develop or derive some mathematization for symmetry in syntax, then one can also develop more rigorous notions of symmetry-breaking in syntax (e.g., for the well-founded formulation that verbs project an asymmetrical relation with external arguments, and many others).

In any system that produces hierarchical embedding the key notion is not symmetry itself, but symmetry-breaking. In linguistic terms, a computable operation Merge will combine two objects $\{\alpha\}$, $\{\beta\}$ to produce a mirror-symmetric object if not restricted in any way. That is, no matter how you flip the two branches of the tree along a vertical axis, the relation will be the same—with the root node label unspecified. Such an object is ambiguous to the extent that the root node label (i.e., projected bundle of lexical features of one or the other $\{\alpha\}$, $\{\beta\}$, or projected Probe of either $\{\alpha\}$, $\{\beta\}$ whose uF gets valued during the merge operation) is ambiguous or unspecified. When no specification for the root node label is given then no hierarchy can be established between the two syntactic objects, as [Boeckx \(2008\)](#) and [Chomsky \(1995c\)](#) discuss. If we iterate this process, unrestricted, then projection of label to the root node will be random, i.e., there is no deciding information for which of the two labels, α or β , projects. Real world syntax necessitates very specific projections—whether they are simple labels, lexical feature bundles, Probe with valued uF , or some other property.⁴ In other words, empirical data show overwhelming evidence that syntactic relations between constituents are not altogether symmetrical and that hierarchical (asymmetric or antisymmetric) relations are a simple fact of human language. The reason for focusing so intently on symmetry, then, is to have some criteria by which to measure symmetry-breaking. Lastly, another benefit of focusing on symmetry is twofold: (i) it has a productive precedence in linguistics with the notion of antisymmetry in ?, and (ii) one need not refer to notions of recursion to talk about symmetry; depending on how this is interpreted, one could say that recursion is byproduct of symmetrical relations. In other words, focusing on symmetry (specifically in a group-theoretic framework) give us recursion for free without having to rely only on recursion. Group theoretic symmetries can

patterns is a sound methodological strategy that unifies linguists in general.

⁴Of course, adjuncts and adjunction present problems of a different sort: such as a possible immediate asymmetry of Merge, represented by the ordered pair set notation $\langle a, b \rangle$ in contrast to typical set notation $\{a\{a, b\}\}$; see [Boeckx \(2008\)](#), [Chamtezky \(2000\)](#) [Hornstein & Nunes \(2008\)](#), and [Rubin \(2002\)](#), [Rubin \(2003\)](#) for examples.

be explained or modeled in many different ways and showing that two elements are symmetric does not necessitate a recursive-generative model, though two symmetric elements can certainly be modeled this way.

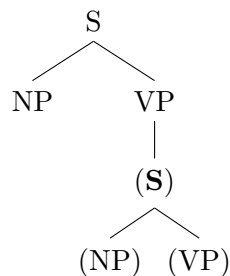
2 α Merge and symmetry

2.1 Ambiguous Merge

Given the naturalistic and rigorous framework for linguistic study briefly discussed in § 1, it is reasonable to experiment with the use of different kinds of rigor. For example, Chomsky (1955) expressed doubts about the kinds of recursive definitions that Bar-Hillel (1954) was talking about, particularly in reference to Carnap (1937). Only later did recursive/computable devices of the form in come to be used.^{5,6}

- (1) General Rewrite Rule with Tree

$S \rightarrow NP VP (S)$



- (2) Other Recursive Rewrite Rules

- a. $NP \rightarrow N (Det) (NP) (PP)$
- b. $PP \rightarrow P (NP)$

Such informal definitions by “recursion” proved to be problematic for describing general environments of linguistic phenomena—to the effect that the definitions in were understood to be too strong (i.e., they produced structures that were either not empirically confirmed or simply ungrammatical). They were quickly generalized under the X-bar schema first proposed in Chomsky (1970) and later extended by Jackendoff (1977).

- (3) X-bar generalization for constituents

- a. $X'' \rightarrow \text{Specifier}, X'$
- b. $X' \rightarrow X, (\text{Complement})$

⁵See § 3.2 for some discussion of the terms “recursive” and “computable.”

⁶Only a few years later Chomsky (1957) would come to recognize, in light of analysis of finite-state grammars expressed in the form of Markov machines, that “If a grammar does not have recursive devices... it will be prohibitively complex. If it does have *recursive devices of some sort*, it will produce an infinitely many sentences” (Chomsky 1957:24, italics mine).

Such that:

- (i). X ranges over N, V, A, P , and S .
- (ii). Endocentricity applies:
 - (ii'). X' is the head of X'' .
 - (ii''). X is the head of X' .
- (iii). Heads share categorial properties with their projections.
- (iv). Complements are X'' categories.

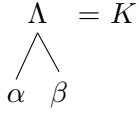
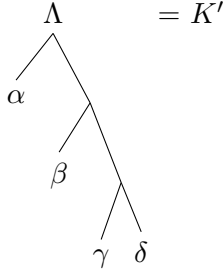
The X-bar model was modified and streamlined by Chomsky (1994) and Chomsky (1995a) in the form of a Bare Phrase Structure (BPS). Since that time, many proposals for a hyper-minimalist BPS have been made on the basis of both theoretical and empirical arguments; for example Carnie (2000), Citko (2005), Collins (2001), and Jayaseelan (2008) to name a few. In BPS, the basic operation Merge *Selects* two syntactic constituents or objects and “merges” them together, giving as its output one constituent or object. The operation itself does not specify which of the two constituent labels projects—projection is assumed to be specified by the features of the objects being merged. For the purposes in this paper, no specification for projection will be given. It is reasonable to call this basic operation “ambiguous,” following Boeckx (2008), and I will denote its operation as αMerge .

Definition 1. $\{\{\alpha\}, \{\beta\}\} \xrightarrow{\text{merge}} \{\Lambda, \{\alpha, \beta\}\} = K = \alpha\text{Merge}^7$

The one thing that all, or most, current BPS models share in common is the general operation Merge defined as a physically computable operation. It has been hypothesized by Hauser et al. (2002) and Fitch et al. (2005) that Merge is operable in the domain of the Narrow Faculty of Language (FLN), specifically Narrow Syntax (NS), and is a unique property of human cognition.⁸ In tree form, αMerge produces

⁷Where Λ = exclusively the label either $\{\alpha\}$ or $\{\beta\}$. Also, where α and β are two independent syntactic objects, K is the new object output of Merge: so $\{\alpha, \beta\} \in K$ (are members of K), but $\{\{\alpha, \beta\}, \Lambda\} \subseteq K$ (both objects *and* the projected label are a subset equivalent to K); see Chomsky (1995c). At first this might appear to violate INCLUSIVENESS, but if one understands that label projection includes only the information contained in the constituents themselves, then a label projection of either α or β seems only to be geometric extension of the constituent information. In essence, the output of merging two constituents becomes the input for a new (automatic) operation of projection—where the feature valuation between the two constituents decides which of them becomes the input to projection. This is a (tail) recursive procedure. It does not violate INCLUSIVENESS. Additionally, it shows that—assuming Merge is a real operation in all human languages, i.e., is part of the language faculty (FL)—recursion is inherent to all human languages whether they show it explicitly or not (e.g., in clause embedding or possessives).

⁸It should be noted that αMerge , as presented here, may be operable in non-human cognition. See Pinker & Jackendoff (2005) for more on this, as well as arguments against Hauser et al. (2002). Also note, αMerge as presented here does not equate directly to adjunction defined in Boeckx (2008), Chapter 3, as a time-dependent insertion $\{\{\alpha\}_t, \{\beta\}_{t+1}\}$ with an inherent temporal asymmetry. Although, there does not appear to be any major incompatibility; especially in light of the fact that my narrowed treatment of Merge is fairly abstract and mathematical in order to get at a group theory definition of symmetry from which more realistic and empirically based definitions of symmetry-breaking can be applied. See § 3.3 for more detail.

Figure 1: α Merge tree produces object K Figure 2: Iterated α Merge tree produces object K'

a mirror symmetric object, seen in Figure 1.

With mirror-symmetric objects each side is the same in terms of their relative position and relation to the other side. Figure 1 has a bilateral mirror-symmetry: rotating the object along a vertical axis 180° or 360° will not change it. Under these conditions it is understood that α Merge is an idealized operation and K is an idealized object. This operation can be iterated infinitely, where each label-node is ambiguous. A finite representation of the iteration is given in Figure 2—notice that K' is a compound of two mirror-symmetric trees but itself has no bilateral symmetry.⁹ Once the bilaterally symmetric object of Figure 1 is repeated to produce a more complicated object it loses its simple mirror-reflection property. By these standards, even if one wanted to argue that α Merge was necessary *and* sufficient to explain why human syntax is the way it is, some notion of symmetry-breaking must be introduced to account for strings containing more than two elements. For example, in Figure 2, $\{\alpha, \beta\}$ dominates the mirror-symmetric constituent $\{\gamma, \delta\}$, even though the label and/or category of $\{\gamma, \delta\}$ is ambiguous. Interestingly, once α Merge projects a label—no matter which label is, for the purposes of this paper, randomly projected from minimal/head level to maximal/phrase level—the number of phrase and head level constituents accumulated as the result of iterated Merge operations correlates with the Fibonacci pattern.

⁹In fact, the tree in Figure 2 could be decomposed into $\{\alpha, \beta\}$, $\{\beta, \gamma\}$ and $\{\gamma, \delta\}$ with 3 ambiguous label slots, including the root node label, and 3×2 possible labels. Generalizing, for n syntactic objects, there are $(n-1)$ merge operations, or $(n-1)$ label slots, and $((n-1) \times 2)$ possible labels. Compare this to the observation in Medeiros (2008) of the relation between *string length* of n and *depth* of $(n-1)$.

2.2 Fib(n) levels and the X-ratio

2.2.1 Rationale and justification

The idealized objects produced in Figures 1 and 2 are problematic because they do not allow *any* specification for label projection. The non-projective model of unrestricted Merge, or α Merge as I am calling it, does not seem to produce any interesting patterns suggestive of natural law. But when one allows α Merge to project a random choice of label the result is an interesting mathematical F pattern, as Medeiros (2008) and Soschen (2008) have shown; see also Boeckx et al. (2006), Carnie & Medeiros (2005), Idsardi (2008), and ?. However, Medeiros (2008:164) points out that

It is tempting to see the appearance of the Fibonacci sequence in the X-bar pattern as being deeply significant in itself. But the X-bar schema is after all a very simple mathematical object, and there may be nothing particularly magical about the appearance of the Fibonacci sequence in the structures it generates. Their appearance in this domain could be no more of a surprise than their appearance in the family trees of bees, or in Fibonacci's idealized rabbit populations, or in the number of metrical possibilities for a line of Sanskrit poetry, or any of the myriad situations these numbers describe. To put it another way, it could be that these properties are an accident of no real significance, or worse, merely a reflection of mathematical simplicity in linguists' description of language, rather than a property of language itself. Yet it is undeniable that patterns related to the Fibonacci sequence play an important role in nature, especially in optimal packing and optimal arboration.

Additionally, Livio (2002) provides analysis for many of the supposed F patterns that have been consciously encoded into art, architecture, and music, concluding that in many of the cases the investigators have been overzealous and mistaken—he even goes so far as to show how the dimensions of his television can be made to fit within the golden mean. Nonetheless, Medeiros (2008) is right, F patterns are undeniable and their investigation in human syntax is significant—if approached soberly.

2.2.2 The golden sequence (GS)

The golden sequence (GS) is derived by a simple algorithm that correlates to the F sequence; it is given in Algorithm 1. A phrase structure equivalent is given in Algorithm 2. The outputs of these algorithms are produced in Appendix A as Table 1 and Table 2—the latter of which generates the tree in Figure 6; see Uriagereka (1998) and ? for linguistic applications of the GS. In the algorithms and their outputs no

attention has been given to bar-level (X') projections; only head (X^0) and phrase (XP) levels are of immediate concern—though Medeiros (2008) and Soschen (2008) include bar-levels. Lastly, the F level refers to the number of syntactic objects (heads and phrases), such that the number of syntactic objects follows the pattern of the F sequence.

Algorithm 1. GOLDEN SEQUENCE:

Start with 1, replace every 1 with 10, and every 0 with 1 in every iterated line.

Algorithm 2. GS FOR PHRASE STRUCTURE:

Start with GS; each 1 = Phrase ($XP = \alpha$); each 0 = Head (X^0).

Importantly, any specific parsing of elements by brackets in Table 2 is not determinate and can be changed. What is important is the fractional relation of heads to phrases, not how these head-phrase relations are parsed.¹⁰ In fact, the parsing, equivalent to the issue of determining the projecting head, is basically random in this approach. Also, it should be obvious that the name of the label does not matter and can be changed.

The GS applied to phrase structure embedding gives similar results to Medeiros (2008), who maps explicit Fibonacci correlates of phrase structure in his approach. His examples (1-5, 2008:153-154) are numbered Phrase Strings 1 to 5 in Appendix B. These Medeiros strings have equivalents when an alternative parsing of the strings from Table 2 is applied—these latter equivalents are numbered Phrase Strings 6 to 10 in Appendix B. Consequently, one can see that the two approaches converge on equivalent results—with obvious and trivial differences in label names; see also Figure 5.

By bracketing the phrase string produced from Fibonacci level 34 (F_{34}) differently than it is parsed in Table 2, a fully balanced binary tree, like the one in Medeiros (2008) and Soschen (2008) can be derived. Figure 3 is an example of a Medeiros tree and Figure 4 shows the same tree without specifiers. The Phrase Strings 11 and 12 represent the latter trees, respectively; 11 for Figure 3 and 12 for Figure 4. Phrase String 13 is derived from the GS algorithm for phrase structure and has been re-bracketed in order to correlate with the previous objects; it represents the embedding for Figure 5. One can easily see that by removing the X' projections from 11, both 12 and 13 are equivalent—and clearly, so are their tree projections.¹¹

¹⁰The F pattern is special because it codes a specific irrational value when two sequential numbers are put into a fraction such that the larger is the numerator; for example 13/8. As the numeral values get larger, and hence the fractions contain larger numeral values, their ratio gets closer to irrational limit. But this is an infinite limit, usually referred to as τ , which has the value 1.618033...

¹¹Potential problems are associated not with the actual re-bracketing, which is not a problem under the assumption of virtually random projections, but of the redistribution of constituents to form the balanced binary tree of Figure 5; compare the right-branching tree of Figure 6 derived from a straightforward sequential embedding of F_{34} .

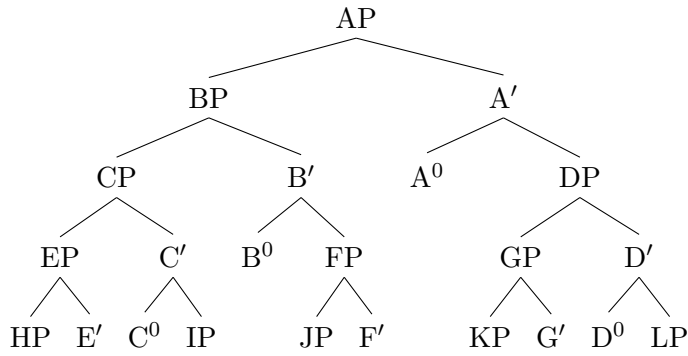


Figure 3: Medeiros tree

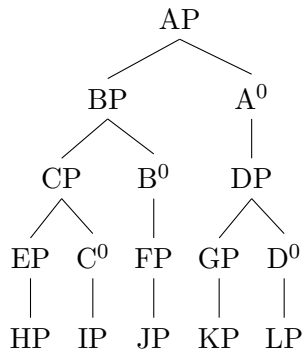
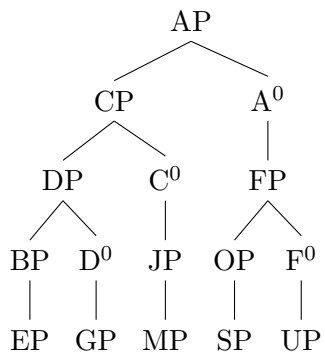


Figure 4: Medeiros tree with no Spec

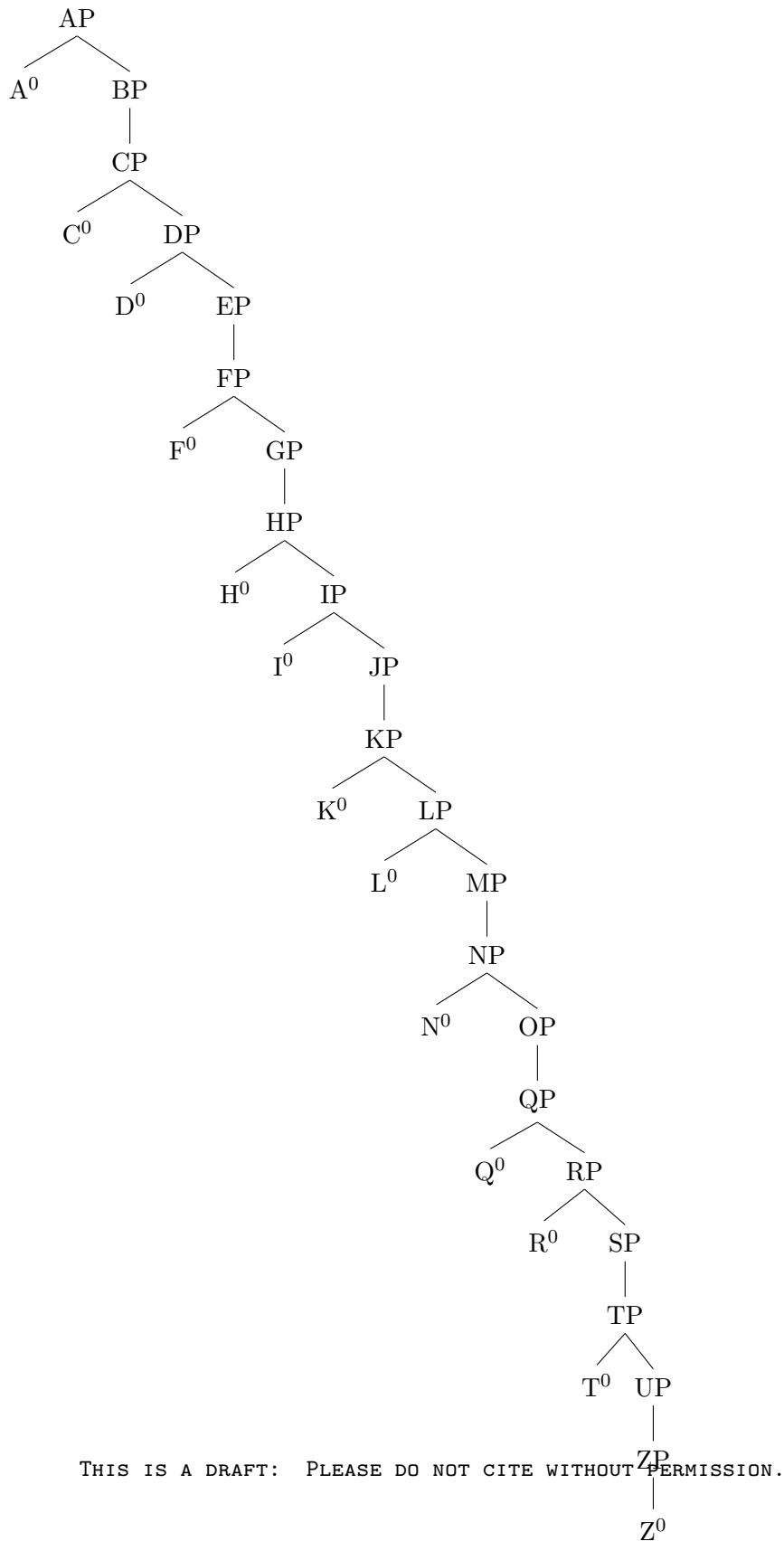
Figure 5: Randomly projecting α Merge tree adapted to Medeiros string

In Figure 6 I produce a binary right-branching alternative to Medeiros’ balanced binary tree.¹² This right-branching alternative is based on a well-known property of F patterns called the Golden Sequence; for example Livio (2002), but especially Uriagereka (1998). It has the same F pattern for syntactic objects that derivations in Medeiros (2008) do, and appears to display the same relations between maximal and minimal levels; though some redistribution of constituents is needed to reformulate the tree in Figure 6 as the Medeiros equivalent in Figure 5. Redistributing constituents in such abstract and idealized models—where label projections are random and no interface conditions or features play a role—does not seem to pose a problem. The placement and order of constituents—basic c-command relations—depends on the order in which they are merged, but in the abstract models used here it does not matter if, for example, BP is merged before AP or I^0 is merged before DP. In other words, if the label projection is random then so is the operation *Select*, which selects items to be merged. What is really at issue here is the fractional relation between the number of constituents at phrase and head levels. Such fractional relations appear to reveal, at a high level of abstraction and idealization, a sequence or pattern that is commonly found in nature: the F pattern/sequence. What is interesting about Medeiros’ treatment of the F pattern and its relation to idealized phrase structure patterns is that—abstracting even more—a $\text{Fib}(n)$ ratio between different comparative levels of phrase structure in a binary tree can be derived. For example, all maximal projections follow an initial sequence of the number pattern, $\text{Fib}(n)$, while terminal heads begin the pattern two levels ‘down’ at $\text{Fib}(n-2)$; (Medeiros 2008: 161). In other words, for maximal projections the sequence begins as shown in (4), but for terminal heads the sequence begins as shown in (5) (again, unlike Medeiros, I ignore intermediate (X') levels here and throughout). It does not matter how one derives the F pattern—whether by the golden sequence or other means—it still produces this ratio.

- (4) Fib sequence for XP: [1, 1, 2, 3, 5, ...]
- (5) Fib sequence for X^0 : [0, 0, 1, 1, 2, 3, 5, ...]

By this relationship between levels, then, the ideal ratio between maximal and minimal projections is $\text{Fib}(n : n-2)$ at any horizontal level—or between sisters—in a binary tree. For example, the $F3$ string in Table 2 has one X^0 and two XPs (A^0 , α , and β) such that $2/1 = 2$. Looking at $F34$, the ratio of XP to X^0 is $21/13 = 1.615384\dots$, an approximate to the limit ratio of the F sequence—usually denoted by ($\tau = 1.6180339\dots$)—for which the ratio of maxima and minima of projections in binary phrase structure trees would get closer as the number of projections reached the limit of infinity within the F pattern. I will refer to the ratio $\text{Fib}(n : n-2)$, defined over maximal and minimal projections in phrase structure, as the X-Ratio; see Carnie & Medeiros (2005), Boeckx et al. (2006), Medeiros (2008), and Soschen

¹²Also work by Alana Soschen, for example Soschen (2008).

Figure 6: Random label projection of α Merge

(2008). It is this ratio in Definition 2 that leads to symmetry and Group Theory.

Definition 2. X-RATIO:

The ideal distributional ratio of maximal projections to minimal projections is $\text{Fib}(n : n-2)$.

2.2.3 Symmetry in syntax

Group Theory is the formal application of study for the investigation of symmetry in nature; see Livio (2005) and Stewart (2007) for popularized accounts and Rosen (1995) and Milne (2008) for formal introductions—also Chapter 10 in Partee et al. (1990). Once the X-ratio has been established, then the symmetry between maxima and minima of constituents, i.e., between phrases and heads, can be defined. By symmetry, I refer to the conjunctive notion defined informally by Rosen (1995).

Definition 3. SYMMETRY:

(i) Possibility of change: It must be possible to perform a change, although the change does not actually have to be performed. (ii) Immunity: Some aspect of the situation would remain unchanged, if the change were performed.

Under this view of symmetry, XP and X^0 are symmetric because the interchange between them of the arithmetical operation $(n \pm 2)$ on the Fibonacci sequence defining them does not change their underlying Fibonacci sequence—it only shifts the X-bar level at which the sequence begins. We might call this a *translational symmetry* such that there is an immunity to change dependent on a certain displacement or shift along a specific boundary. Such *translational* symmetries produce repeating patterns. Another way to describe the X-ratio would be as a *level symmetry*: XP and X^0 are immune to change—in terms of their Fibonacci patterns—relative to minimal or maximal level; i.e., relative to their head or phrase level. The difference, then, between maximal and minimal projections in binary trees is $\text{Fib}(n \pm 2)$, which determines the X-bar level at which the F pattern begins, as in (4) and (5). This is generally consistent with the empirical and theoretical claims of the hyper-minimalist BPS of, for example, Carnie (2000), where lexical features can be projected as XP or X^0 depending on specifics of the syntax of predication.

However, the symmetry noted here is not the symmetry of α Merge in § 2.1. The X-ratio suggests a symmetry developed from constraints that determine the label projection; albeit the constraints used here are practically random. Needless to say, empirical data is more complicated and provides non-random constraints for the specification of projection. Importantly, the symmetries noted here and in § 2.1 do share one thing in common: they are both symmetries of syntactic objects and not of, say, syntactic rules of the sort discussed in Boeckx (2008).¹³

¹³For example, the “Merge is source-independent” rule which applies to a symmetry between External and Internal Merge such that under the operation *Select*, both types of Merge are immune to change; i.e., Merge is the same operation in all contexts. As a sidenote, Boeckx (2008) highlights

Thus, heads and phrases are symmetrical in an important way that differs from traditional notions of symmetry in syntax.¹⁴ The current status of the notion of symmetry in syntax and prosody—based on empirical and theoretical observations—is suggestive of possible new directions and certainly justifies a consideration of GT and anything this particular formalism might offer as useful tools to the linguist. I suspect that what is developing in current theoretical research is the groundwork for a potential discovery of a general principle (or group of principles) from which head and phrase levels can be derived. That is, head and phrase levels no longer need to be thought of as syntactic or linguistic primitives. Instead, they may be products of a general principle(s) of phrase structure based on some first principle(s) of natural law for efficient self-organization.¹⁵ These laws or first principles may be coupled with a combination of interface constraints and/or syntactic operations, such as Agree in Boeckx (2008), that essentially work to break the initial symmetry of ambiguously merged elements or elements that have the potential for head-phrase (X-ratio) ambiguity.¹⁶

Of some immediate relevance here is the role of the formalisms of GT—the mathematical system used to study symmetry. It may not be necessary to incorporate much of the formal mechanism of GT, certainly not all of it, just as it is not necessary to incorporate most or all of Computability Theory simply because the notion of computability is crucial to theoretical syntax. However, having said this, it is at the very least interesting to flesh out some basic observations of symmetry in syntax in terms of the GT formalism; but I will not go into great detail.

2.2.4 A conceptual rationalization for GT and symmetry

The benefit of GT is that it can model practically any kind of real world object (animate bodies, the Rubik’s cube) and it can model abstract laws or rules. Boeckx (2008) has shown how a rotational symmetry can be used to model symmetrical relations between projections and chains (pg. 46), see also Uriagereka (1998), and he has also modeled symmetry relations between specific linguistic rules to point out the essential fractal (i.e., self-repeating patterns at different levels of scale) nature of syntax (pg. 159); see also Boeckx & Uriagereka (2007) for discussion of fractals in nature and the Minimalist Program. Medeiros (2008), Soschen (2008),

how this observation can be traced back to the *Structure Preservation Hypothesis* of Emonds. This seems like a significant advance in syntactic theory.

¹⁴Such as subject-object asymmetries.

¹⁵This kind of thinking meshes very well with Baker (2003) and his attempt to derive category information from more basic syntactic structural relations. Baker defines two such relations in the following way: *X is a Verb (defined in 1) or a Noun (defined in 2) if and only if X is a lexical category and X*

1. has a specifier.
2. bears a referential index, expressed as an ordered pair of integers.

¹⁶For example, clitics or nominal predicates.

Idsardi (2008), and ? have pointed out relevant patterns of F sequences in idealistic models of prosody, X-bar structure, and phase alternations. F sequences can be defined as a type of self-similar iteration that has a fractional limit of the value τ ($= 1.618033\dots$).¹⁷ This value is also the ratio responsible for making symmetrical spirals and other types of symmetrical objects. If one takes seriously the notion that the human ability to acquire and use a linguistic system—a system unique in the animal kingdom—is really more of a reflex to a natural process based on a “growth form,” then the idea that symmetry (and more importantly symmetry-breaking) is a necessary property of such a “growth form” is a tantalizing possibility. Symmetry is one of the most profound properties of nature—and in fact, it is the slight deviation from symmetry that is so profound. It means that multiple types of natural objects constructed of multiple kinds of material are at some abstract level following a similar blueprint. From spiral galaxies to human faces, this abstract blueprint is a defining characteristic of the laws of nature (not to mention fractals, which symmetrical forms can be related to). There is no reason to deny that, at an appropriate scale of measurement and abstraction, the “growth form” of the human language system must reveal the property of symmetry—and in fact does show deviance from symmetry in its real-world form.

3 Strong Minimalist Thesis

This section covers three areas that form part of the backdrop for evaluating directions and formalisms that the MP can utilize—they are indirectly related to the “three factors” of Chomsky (2005). The first area deals with the biological nature of natural language. This means that although operations like Merge can be defined by rigorous formal and mathematical tools, Merge itself is not a formal object. Furthermore, the fact that an interesting mathematical pattern found throughout nature can also be found in prosodic and phrase structures does not mean that these structures are themselves mathematical or formal objects. (In fact, it does not even mean that the mathematical patterns are *inherent* to prosody or X-bar; the pattern may simply be a matter of our perception). The Strong Minimalist Thesis (SMT) proposes a general methodological backdrop for this and states, generally, that language is an optimal form meeting interface conditions. This implies that natural language is not a formal object, but an object of nature. The next area deals with the general notion that we expect, in lieu of strong contrary empirical evidence, operations in language to be in fact *operations that are computable*, i.e., specifiable by an algorithm that can be discovered or constructed. These computable operations must be physically tractable. That is, operations are bounded by human processing limits. Therefore, MP theories about syntax must meet certain natural limits. The

¹⁷When one makes a fraction of any two sequential values in the F sequence, say $[\dots, 5, 8, \dots]$ as $8/5 = 1.6$ then take the next two values $[\dots, 8, 13, \dots]$ as $13/8 = 1.625$, one always gets closer to the limit value of τ but can never reach the limit because it is infinite.

third area is about the nature and concept of computability itself and is intimately related to the second area. It deals with what linguists mean by the terms “computable” and “recursive.” The following subsections briefly outline these areas in order to give an explicit context for evaluating the usefulness of GT and formal notions of symmetry in syntax.

3.1 Merge is not a formal object

Natural language is not a formal object. In this regard, neither is the operation Merge—though it, and its historical antecedents, are regularly defined through formal mathematical means. As early as Chomsky (1955) and Chomsky (1957) the status of natural language as a formal object has been viewed critically within the context of twentieth-century advances in logical syntax and semantics of the type found in Carnap (1937), Ayer (1936), Reichenbach (1947), Quine (1953), and Tarski (1956). This does not mean that useful approaches from formal theories cannot be used productively, only that natural language probably cannot be modeled *exhaustively* by any formal/mathematical system that yet exists—the implication here is that it is up to linguists to construct such a system from the formalisms that do exist, coupled with empirical evidence from natural languages; for example Langendoen & Postal (1984), Partee et al. (1990), Hornstein (1992), Prince & Smolensky (1993), Uriagereka (1998), Chierchia & McConnell-Ginet (2000), Tesar & Smolensky (2000), Bod et al. (2003), Niyogi (2006), and of particular relevance here Fortuny & Corominas-Mutra (to appear). No matter the formal methods that are drawn on (usually set theory in semantics and syntax and probability theory in phonology and phonetics) it has generally been understood that, as (Fortuny & Corominas-Mutra to appear: 7-8) state,

As any abstract theoretical background, it is not reasonable to ask about the reality of the operations and objects defined. For example, although the algorithm runs through a time step indicator, such time step is only given for operational purposes and does not imply—in our field of study—any temporal evolution. What is reasonable to ask is whether from the defined mathematical framework we can derive the core properties that we observe in the studied object (pp. 7-8).

(Chomsky 1955: 45) expressed a similar attitude in response to arguments by Bar-Hillel (1954) for the use of formal models in natural language—specifically the use of recursive definitions in a formal syntax of the type found in Carnap (1937):¹⁸

¹⁸Interestingly, as Tomalin (2006) points out, Bar-Hillel’s linguistic work served as the foundation for a specific approach to natural language, Combinatorial Categorical Grammar (CCG) widely used in computational linguistics.

[t]he correct way to use the insights and techniques of logic is in formulating a general theory of linguistic structure. But this fact does not tell us what sort of systems form the subject matter for linguistics, or how the linguist may find it profitable to describe them. *To apply logic in constructing a clear and rigorous theory is different from expecting logic or any other formal system to be a model for linguistic behavior (italics mine).*

Additionally, a realistic theory of linguistics that takes natural language as a scientific object of inquiry must subject the brain to the known laws of biology, physics, and nature in general. In computational complexity theory, for example Deutsch (1985), Deutsch (1997), Lloyd (2000), Lloyd (2006), and also Kauffman (1995), all physical systems are understood to be computable and subject to the known laws of physics and computation, specifically entropy (the second law of thermodynamics) and universal computability (the Church-Turing Thesis). By normative scientific standards the brain, and by implication FL, cannot contradict laws of nature. I explicitly state this as a corollary of the *weak minimalist thesis*, which is an uncontroversial weakening of the strong minimalist thesis proposed in Chomsky (2008).¹⁹

Strong Minimalist Thesis. Human language is an optimal solution to interface conditions that the faculty of language (FL) must satisfy.

Proposition 1. *Weak Minimalist Thesis:* *Human language is the natural product of linking sound (or gesture) and meaning.*

Corollary to Weak Minimalist Thesis. Human language is a product of the natural state of the organism-as-a-whole, and therefore, is subject to the same laws and constants that all of nature is.

Notice that the Corollary to the WMT does not say that human language, or more precisely FL, is explained by or reduced to all natural laws; nor does it say that all natural laws apply to FL—only that FL is subject to the laws and constants of nature (and by implication only a subset of natural laws). The Corollary is simply a compression of arguments found in many sources, including Boeckx & Piattelli-Palmarini (2005), Chomsky (1986, 1995b,c), ?, Uriagereka (1998) and many others past and present; see also Formigari (2004) for historical consideration of issues relevant to the Corollary—particularly historical tensions between assigning natural language a “spiritual” or “bestial” cause relative to “immaterial” and “material” effects.

¹⁹The SMT is widely regarded as not being plausible—though possible—and is thus used as a heuristic standard. For this reason, it seems strange to derive a corollary from a heuristic. This is the motivation for putting forward a weakened version of the SMT.

3.1.1 Assumptions and the Church-Turing Principle

Assume the faculty of language (FL) is a finitely realizable physical system with an initial state Σ_{UG} of Universal Grammar followed by subsequent computational states (C_1, \dots, C_n) . These states have a finite limit simply because they emerge from and are 'computed' by a finitely organic brain—a product of biological nature. But a finite realistic limit does not negate the notion of *discrete infinity*. The latter is a property of the computability of FL, not its real-world limit. In theory the initial state of FL, Σ_{UG} , has the computable potential of reaching infinity, Σ_∞ , but because of the inherent constraints of natural law applied to biological systems no real (i.e., physically computable) FL actually reaches the limit. Further assume, as any physical system that can be realized by computable functions, the FL can be modeled by a universal Turing machine, (*uTM*), as determined by the informal definition of the Church-Turing Thesis (CTT) given by Deutsch (1985). However, Deutsch (1985) rejects the CTT as too vague and opts for a redefinition he calls the Church-Turing Principle (CTP). Assume the CTP applies to the MP.

Church-Turing Thesis. Every function which would naturally be regarded as computable can be computed by the universal Turing machine.

Church-Turing Principle. Every finitely realizable physical system can be perfectly simulated by a universal model computing machine operating by finite means.

(Deutsch 1985: 100) elaborates:

I propose to reinterpret Turing's "function which would naturally be regarded as computable" as the functions which may in principle be computed by a real physical system. For it would surely be hard to regard a function 'naturally' as computable if it could not be computed in Nature, and conversely. To this end I shall define the notion of 'perfect simulation'. A computing machine M is capable of perfectly simulating a physical system S , under a given labelling of their inputs and outputs, if there exists a program $\pi(S)$ for M that renders M computationally equivalent to S under that labelling. In other words, $\pi(S)$ converts M into a 'black box' functionally indistinguishable from S

A syntax utilizing the formalization of a *uTM* in the linguistic domain has been extremely successful, especially if viewed from the perspective of the CTP. According to the MP, the physical system S of the CTP is the brain—and there is an expectation that the computability of it be physically tractable, at least in terms of the computability of generating hierarchically nested structures that can be iterated (in)finitely to produce various forms (i.e., discrete infinity). For example, Chomsky (2005) states the third factor of language design as "Principles not specific to the faculty of language. . . including principles of efficient computation" (pg.

6). Here, efficiency is relative to the naturally finite biological limits of the human brain. There is a strong (conceptual) parallelism between Deutsch’s redefining CTT as the physically bounded CTP and MP interests in the physical tractability of computable operations like Merge. Of course, David Deutsch employed his redefinition in terms of quantum phenomena and the computability of the universe—theoretical linguist’s concerns for physical computability are dramatically narrower and more deterministic. However, this should not restrain linguistic interest in the physical computability of nature as long as one can see that such parallels here are, at first glance, simply analogous, conceptual, and intended to inspire—not direct a path of research.

3.2 Normative use of the term “computability”

Although the normative use of term “computability” may seem like a mundane topic, it in fact strikes to the heart of the issue of what linguists mean when they use the term “recursive.”²⁰ Soare (1996), and more recently, Soare (2007, 2008), has pointed out a historical confusion in the use of the two terms that may lead to unfortunate misunderstandings relative to the field of study and the definitions

²⁰One example of “recursive” can be found in (Partee et al. 1990: 181-82), who show that a recursive definition can be gotten by considering

the set M of all mirror-image strings on $\{a,b\}$. A mirror-image string is one that can be divided into halves, the right half consisting of the same sequence of symbols as the left half but in the reverse order. For example, *aaaa*, *abba*, *babbab*, and *bbabbabb*... The following is a possible recursive definition of M .

1. $aa \in M$ & $bb \in M$
2. $(\forall x)(x \in M \rightarrow (axa \in M \text{ \& } bxb \in M))$
3. M contains nothing but those members it has by virtue of lines 1 and 2

Line 1 is the base, line 2 is the recursion step, and line 3 is the restriction. They go on to define the Principle of Mathematical Induction:

For any predicate Q , if the following statements [1 and 2] are both true of Q then the following statement [3] is also true of Q :

1. $Q0$
2. $(\forall x)(Qx \rightarrow Q(S(x)))$
3. $(\forall x)Qx$
(pg. 196)

Where $S(x)$ is the successor function—denotes the successor of (x) —and $Q0$ denotes the property Q of zero. They note that the similarity between lines 1 and 2 in both demonstrations is “readily apparent.” But also say that the “Principle of Mathematical Induction is not a definition, however, but a rule of inference to be applied to statements about the integers” (pg. 196). Whatever similarities exist, recursion and induction as given here are not, arguably, equivalent in the strong sense. Furthermore, even if they are equivalent in the weakest sense of the term, the applications to which mathematical induction is relevant are not the kinds of applications relevant to the physically tractable computability concerns of the linguist.

assumed. He points out that there are both technical and now normative differences between the terms (i) “(primitive) recursive function” and (ii) “recursively enumerable” (r.e.). The latter term is closer to what linguists know as “recursion,” and as Soare suggests, there are important reasons for clearly delimiting the difference in terminology. Substituting (ii) with the term “computable” and making it clearly different from “recursion” will have positive results in the sciences. (Soare 1996: 34) states that anyone using the term “recursive” always risks the chance that a mathematician or computer scientist will confuse it with induction: “when we use ‘recursive’ to mean ‘computable,’ we are using it in a way that is not in any dictionary and which an educated scholar or scientist cannot reasonably be expected to know. Indeed, there is a danger that a computer scientist or mathematician might mistake it for ‘inductive.’” I use the term Turing computable (TC) or “computability” to designate r.e., or the linguistic notion of “recursion.” Additionally, while discussing the fact that Emil Post’s work focused directly on computably enumerable sets (which are usually called recursively enumerable sets) instead of computable functions—as found in the work of Church (1936) and Turing (1936), although Post (1936)—(Soare 2008: 25) says that

concentration on c.e. [computably enumerable] sets rather than partial computable functions may be even more fundamental than the thesis of Church and Turing characterizing computable functions because... often in higher computability theory it is more convenient to take the notion of a generalized c.e. set as basic and to derive generalized computable functions as those whose graphs are generalized computably enumerable.

Recall that (Chomsky 1965: 9) partly attributes the use of the term “generative” to the work of Emil Post (Chomsky is defending the normative and technical use of “generative”). A relevant quote can be taken from (Post 1944: 286), also in Soare (2008): “every *generated set* is effectively enumerable, every effectively enumerable set of positive integers is recursively enumerable” (*italics mine*). Chomsky attributes part of the motivation for the term “generative” to Post’s work on—what Soare argues should now be called—computably enumerable sets. That is, “generative” seems to refer more to functions defined by (or sets enumerated by) Turing machines, algorithms, and concepts generally associated with computation than to definition by induction, general recursive functions defined by Herbrand-Gödel, or fixed points in the Kleene Recursion Theorem. These latter notions do not generally appear to be utilized by linguists and are not familiar as part of the formal methods used to investigate I-language. Of the latter terms specific to recursion there is one possible exception: an apparent equivalence between **definition by induction** and **definition by recursion**, but this equivalence is only arguably apparent—though I give no demonstration. Additionally, even if a significantly relevant equivalence between definition by recursion and definition by induction is real,

this does not change the fact that linguists do not use the latter because it is in fact only useful for investigating the natural numbers \mathbb{N} and arithmetic in general—not natural language syntax and semantics. Nonetheless, the concept of definition by recursion has been used widely; example in this paper, [Partee et al. \(1990\)](#), and [Chierchia & McConnell-Ginet \(2000\)](#).²¹ But this kind of definition, used within the context of linguistics, is useful only in order to construct algorithms that are (finitely) computably efficient and physically tractable within the general domain of human cognition—and in terms of the operation Merge, within the domain of narrow syntax (NS) and the narrow faculty of language (FLN). Lastly, ([Soare 1996: 29](#)) proposes that a historically normative convention has largely directed the use, and sometimes misuse, of the term “recursive.”

The Recursion Convention has brought ‘recursive’ to have at least four different meanings.... This leads to some ambiguity. When a speaker uses the word ‘recursive’ before a general audience, does he mean ‘defined by induction,’ ‘related to fixed points and reflexive program calls,’ or does he mean ‘computable?’ [...] Worse still, the Convention leads to imprecise thinking about the basic concepts of the subject; the term ‘recursion’ is often used when the concept of ‘computability’ is meant. (By the term ‘recursive function’ does the writer mean ‘inductively defined function’ or ‘computable function?’) Furthermore, ambiguous and little recognized terms and imprecise thinking lead to poor communication both within the subject and to outsiders, which leads to isolation and lack of progress within the subject, since progress in science depends on the collaboration of many minds.

Group Theory is not so much explicitly employed in computability theory but set theory is, and since GT presupposes some set theory it is an important, albeit minor, issue to note that research into how the operation Merge works is research into its computable nature—NOT its recursive nature. As with the parallel to the CTP, linguistic science also has parallels with computability theory and shares the latter’s distinction between (i) an operation M being computable by finite means that are describable by algorithms—for example [Fortuny & Corominas-Mutra \(to appear\)](#)

²¹[Chierchia & McConnell-Ginet \(2000\)](#) state for the bracket $[_A B C]$ that

We have to specify the value of the tree whose root is A in terms of the values of the subtrees rooted in B and C . This means that the semantic value for the terminal string dominated by A is determined in terms of the values of the substrings dominated by B and C and the way these substrings are put together. If we do this for every syntactic rule in the grammar, we can interpret any tree admitted by it. A definition of this kind (with a finite number of base clauses and a finite number of clauses that build on the base clauses) is called *recursive* (pg. 76).

are developing a nesting machine based on a set-theoretically defined algorithm within the general theory of order—or trying to determine computable operations for certain functions of M , or even generating sets of objects that are countable by M , and (ii) an operation M' that is defined by induction or uses primitive recursive functions. It is within the general backdrop of computability, not recursive function theory, that any formalism of GT will be successful in theoretical syntax. An analogy can be made to the linguistic use of Turing machines (TM): theoretical syntacticians are more concerned with a specific type of universal TM²² as it relates to the operation Merge within the larger backdrop of the concern for algorithms that are both physically tractable and computably efficient.

3.3 GT and syntax

Group Theory, or GT, presupposes some knowledge of set theory because, technically, any group is also a set. That is, any collection of objects, including a set or collection of sets, can be a group if it can be defined by Definition 4; the following is loosely based on Rosen (1995).²³

Definition 4. GROUP:

G is a group iff G, under a law of composition, meets Criterion 1 to 4.

Criterion 1. Closure:

For all $a, b : a, b \in G$, then $ab, ba \in G$.

Criterion 2. Associativity:

For all $a, b, c : a, b, c \in G$, then $a(bc) = (ab)c$.

Criterion 3. Existence of Identity:

G contains identity element $e : ae = ea = a$.

Criterion 4. Existence of Inverses:

For every $a \in G$, then $a^{-1} \in G : aa^{-1} = a^{-1}a = e$.

A “law of composition” is broadly defined as any procedure that combines any two elements in any way (i.e., a binary operation). This can include the combination of functions $f(n)$, relations R , integer values in addition or multiplication ($t + y$, $t \times y$), or the combination of collections of objects that are subsets of a set $s \in S$. Compositions also include permutations, rotations, and translations by displacement of a certain distance along a certain line. (For group theory, see especially Chapter

²²This is the Turing *automatic*-Machine, compared to a stochastic/probabilistic TM, or even a quantum TM that Deutsch (1985) formalized.

²³Notice an interesting parallel with the criterion given below and some examples from Boeckx (2008) when discussing the “symmetry problem” for Merge (pp. 79-80). In discussing the empirical inadequacy of, what I am calling α Merge, he stresses that it cannot produce *asymmetric* relations such as precedence ($a \prec b \neq b \prec a$), and also prosody defined over prominence ($[[A, B] C] \neq [A [B, C]]$). These are equivalent to Criterion 1 and Criterion 2, respectively.

10 in Partee et al. (1990); also Milne (2008), section 39 in Kleene (2002 [1967]); also Korfhage (1974), Dornhoff & Hohn (1978)—or any abstract or modern algebra book; also Livio (2005) and Stewart (2007) for historical and popularized introductions to symmetry and groups.) The genius of GT is that it is so broadly defined that it can describe a wide variety of concrete or abstract objects, but is rigorous enough to derive very precise relations between those objects. I begin with a general notion for a law of composition.

Law of Composition. (informal):

Any two possible elements, a , b , may be combined in any possible way, where ‘ \circ ’ is any combination, iff $a \circ b$ equals a set S .

I now briefly sketch how α Merge meets the criteria in Definition 4, giving a proof. I then discuss some issues arising from the conjecture that syntactic operations can be defined using criteria from GT.

Conjecture 1. The operation α Merge forms a group.

Proof. Assume no other operation Merge and no interface constraints, then the following conditions hold:

Closure: If $\{\{\alpha\}, \{\beta\}\} \xrightarrow{\text{merge}} \{\Lambda, \{\alpha, \beta\}\} = K$, (where $\Lambda = \alpha$ or β), then $\{\alpha, \beta\} \in K$ and $\{\beta, \alpha\} \in K$. This satisfies closure and is analogous to the c-command relation R in the following way: if $\alpha R \beta$, then $\beta R \alpha$.

Associativity: If $[\Lambda \alpha [\beta [\gamma \delta]]] = K'$ and all label projections are ambiguous, then K' equals the decomposition $\{\alpha, \beta\}$, $\{\beta, \gamma\}$ and $\{\gamma, \delta\}$. By associativity, then, K' can also be decomposed into $\alpha\{\beta, \gamma\}$ and $\{\alpha, \beta\}\gamma$. This is analogous to the c-command relation R in following way: if, for example, $\alpha R \{\beta, \gamma\}$, then $\{\alpha, \beta\} R \gamma$.

Identity: By analogy, where rotation by 360° is equivalent to the absence of rotation—which is equivalent to an identity for rotation—the absence of the operation α Merge occurring is equivalent to the identity function. Then, by stipulation $\exists(\neg\alpha\text{Merge})$, and by identity, $\{\alpha\}$, $\{Y\} \xrightarrow{\neg\text{merge}} \{\alpha\}$, where Y is any constituent or set of constituents.

Inverse: By stipulation, if $\exists\{\alpha\}$ then $\exists\{\alpha^{-1}\}$.

□

The last stipulation for the existence of inverses is the biggest problem. Compared to the stipulation for the existence of identity—which is simply the non-occurrence of the operation α Merge—the existence of inverses is harder to rationalize. Because it is specific for actual constituents in the derivation it might be interpreted as saying that lexical or functional items have inverses—or that lexical entries must encode some kind of information for its inverse. This is not so; in fact, the inverse can be part of the mechanism of a numeration N and not of actual constituents. For now, I assume the legitimacy of this reasoning based on the following line of argument. Assume a numeration N consists of a finite set of elements

that must all be selected, merged, and meet further requirements specific to Case, Phase, Agree, and other mechanisms or features belonging to a real-world syntactic derivation. Within a particular N , all elements consist of a set of pairs (LI, i) where LI is the lexical item and i is the index, Chomsky (1995c). Once the index i is reduced to zero, such that it can no longer be selected to merge by External Merge (an extension or equivalent of α Merge), then it seems rational to conclude that such a zero index specification may act as the inverse for whatever LI it was indexing. For items with an integer value greater than 1 for its index—so that it must be selected multiple times in order to reduce to zero—the inverse is formed from its correlational index. That is, $(DP_1, DP_2, \dots, DP_n)$ correlates to $(DP_{-1}, DP_{-2}, \dots, DP_{-n})$. In this way no problem arises for having to propose an inverse for lexical items. Instead, the inverse belongs to the abstract structure of the numeration. In other words, inverses are constructed from the indices of N so that an item in the numeration with (LI, i) as its pair will also have (i^{-1}) and would look like this: (LI, ii^{-1}) . This is an explicit representation of the idea that an index *must* reduce to zero (actually its inverse). Under this interpretation then, indices do not actually reduce to a numerical zero, but instead, they reduce to their prespecified inverse. Consequently, since the index is really just a placeholder within the numeration and does not belong to the feature content of LI 's, the inverse of this placeholder is also not specified as a feature of LI , but of the abstract structure of N generally.

What GT has to offer syntax generally deals with the nature of derivations. For example, trees derived by iterations of α Merge, such as Figure 2, can be shown to form an Abelian group because any element can be *commuted* with any other element. This is certainly not the case with real-world trees derived from a more complicated and natural operation Merge. However, some elements in trees can commute with each other. Or rather, some *positions* can commute based on feature specification and probe-goal relations. These form chains CH , which, under GT considerations, might be Abelian subgroups g belonging to a non-Abelian group G . Certainly, in GT these kinds of things are possible. And if the group G is redefined as the derivation D , then a chain CH of D may possibly form such a subgroup relation.

Applications from a GT perspective, namely one that defines α Merge by Criterion 1 through 4, remain to prove their worth—but the possibility of rigor for defining a base notion of symmetry from which Merge originates and acts as *symmetry breaking agent* is promising. Such possibilities for syntax make certain aspects of GT a welcome convenience. It may provide formal rigor in establishing interesting relations of general symmetry, and especially symmetry-breaking, in natural language.

Conclusion

This paper has given a very brief historical background to the operation Merge, showing how its current conception is the result of a narrowed focus on the role of the logical tool of *definition by recursion* for constituents. Early transformational-generative grammars specified unique recursive definitions for constituents, example , while the X-bar model that followed generalized this definition in the form of a variable defined over lexical and functional categories, example (3). The current MP notion of Merge is that it is an abstract operation occurring only at local intervals, blind to future operations, and based strongly on a Bare Phrase model of the X-bar type. Given this context, Medeiros (2008) has constructed an idealized balanced binary branching tree that exhibits a common mathematical pattern found in nature, the Fibonacci pattern, that correlates the number of X-bar objects/levels with the mathematical pattern that defines a Fibonacci level in a downward sequence; see also Uriagereka (1998), Carnie & Medeiros (2005), Boeckx et al. (2006), Idsardi (2008), Soschen (2008), and ? for use of the F-pattern. I show that if one follows the algorithm of the Golden Sequence, which derives the F-pattern, and correlates the substitution criteria (1, and 0) with XP and X^0 levels, a right-branching equivalent to Medeiros (2008) and Soschen (2008) results. By abstracting further, and looking only at head and phrase levels, a *translational symmetry* can be seen between the two levels and defined as a ratio between their co-ordinate F-levels as $F(n : n - 2)$; also called the X-ratio. This is one measure of symmetry that can be found in syntax, but there are other non-trivial kinds, as Boeckx (2008) shows.

This paper also discusses issues relevant to methodological concerns stemming from the Strong Minimalist Thesis as they relate generally to the potential use of formal tools in syntax, namely tools from Group Theory. These issues deal mainly with (i) the biological nature of natural language, (ii) that operations in natural language are in fact *operations that are computable*, and (iii) the nature and concept of computability itself. An argument for the biological nature of Merge is given, contrasted to a view of Merge—and human language—as a formal object. Additionally, a weakened version of the SMT is proposed and a corollary derived from it—both focusing on the view that human language is at least partially a product of nature. Working from the “Corollary to the Weak Minimalist Thesis,” I highlight some parallels between the MP and research into concepts of the computability of natural systems—though such parallels are meant to be taken in a conceptual and analogous light. Most useful in this context is the Church-Turing Principle, which is a naturalistic redefinition of the Church-Turing Thesis, given in Deutsch (1985). Following this, I focus on arguments made by the mathematician and computability theorist Soare (1996, 2007, 2008) about the normative use of the term “computability,” in contrast to the sometimes misused term “recursive,” as it applies to the linguistic field. It seems reasonable that, given linguists are more concerned with algorithms computable through discretely infinite means and processed by finite

machines (brains), the use of the term “recursive” should be dropped in favor of “computability;” the latter of which already implies the kind of recursivity linguists are interested in.

Finally, the fact that various kinds of non-trivial natural symmetries can be defined in syntax is strong support for the investigation of the formal study of symmetry, Group Theory. Additionally, given the far-reaching goal for more rigorous methods of measurement and analysis in syntax, it is reasonable to look at Group Theory as possibly providing some useful tools to the theoretical linguist. For this reason, I provide a group theoretic definition of the idealized operation ambiguous Merge, α Merge, and briefly discuss some implications of this definition—specifically that a numeration N may have an inverse i^{-1} for any index i that exists. Now, instead of an index reducing to zero, it simply reduces to its inverse. There also appear to be, on the face of it, possible uses of Group Theory in defining, measuring, or analyzing the behavior of Chains and their formal relations to derivations.

Appendix B

Phrase String 1. $[\alpha [\beta \dots [\gamma [X^0 \delta]] \dots]]$

Phrase String 2. $[X^0 \alpha]$

Phrase String 3. $[\alpha [X^0 \beta]]$

Phrase String 4. $[\alpha [\beta [X^0 \gamma]]]$

Phrase String 5. $[X^0 [\alpha\beta]]$

Phrase String 6. $[\beta [\gamma \dots [\delta [C^0 \epsilon]] \dots]]$

Phrase String 7. $[A^0 \beta]$

Phrase String 8. $[\alpha [A^0 \beta]]$

Phrase String 9. $[\beta [\gamma [B^0 \delta]]]$

Phrase String 10. $[A^0 [\beta\gamma]]$

Phrase String 11. $[AP [BP [CP [EP HP E'] [C' C^0 IP]] [B' b^0 [FP JP F']]] [A' A^0 [DP [GP KP G']]$

$[D' D^0 LP]]]$

Phrase String 12. $[AP [BP [CP [EP HP][C^0 IP]] [B^0 [FP JP]]] [A^0 [DP [GP KP][D^0 LP]]]$

Phrase String 13. $[AP [CP [DP [BP EP] [D^0 GP]] [C^0 [JP MP]]] [A^0 [FP [OP SP] [F^0 UP]]]$

References

- Ayer, A. J. 1936. *Truth, logic, and Language*. New York: Basic Books.
- Baker, Mark. 2003. *Lexical categories*. Cambridge, UK: Cambridge University Press.
- Bar-Hillel, Yehoshua. 1954. Logical syntax and semantics. *Language* 30(2): 230–237.
- Bod, Rens, Jennifer Hay & Stefanie Jannedy (eds.). 2003. *Probabilistic linguistics*. Cambridge, MA.: MIT Press.
- Boeckx, Cedric. 2008. *Bare syntax*. Oxford: Oxford University Press.
- Boeckx, Cedric, Andrew Carnie & David Medeiros. 2006. Some reflexes of natural law in syntactic structure. Ms., Harvard University and University of Arizona.
- Boeckx, Cedric & Massimo Piattelli-Palmarini. 2005. Language as a natural object—linguistics as a natural science. *The Linguistic Review* 22: 447–466.
- Boeckx, Cedric & Juan Uriagereka. 2007. Minimalism. In Gillian Ramchand & C. Reiss (eds.) *Oxford handbook of linguistic interfaces*, 541–573. Oxford: Oxford University Press.
- Carnap, Rudolf. 1937. *The logical syntax of language*. Chicago: Open Court Classics.
- Carnie, Andrew. 2000. On the definition of X^0 and XP. *Syntax* 3(2): 59–106.
- Carnie, Andrew & David Medeiros. 2005. Tree maximization and the extended projection principle. *Coyote Working Papers in Linguistics* 14: 51–55.
- Chamtezky, Robert. 2000. *Phrase Structure: From GB to Minimalism*. Oxford: Blackwell.
- Chierchia, Gennaro & Sally McConnell-Ginet. 2000. *Meaning and Grammar: An introduction to Semantics*. Cambridge, MA.: MIT Press, 2nd edn.
- Chomsky, Noam. 1955. Logical syntax and semantics: Their linguistic relevance. *Language* 31(1): 36–45.
- Chomsky, Noam. 1957. *Syntactic structures*. The Hague: Mouton.
- Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge, MA.: MIT Press.
- Chomsky, Noam. 1970. Remarks on nominalization. In Roderick Jacobs & Peter Rosenbaum (eds.) *Readings in Transformational Grammar*. Waltham, MA.: Ginn.
- Chomsky, Noam. 1986. *Knowledge of language: Its nature, origin, and use*. New York: Praeger.
- Chomsky, Noam. 1994. Bare phrase structure. *MIT Occasional Papers in Linguistics* 5.
- Chomsky, Noam. 1995a. Bare phrase structure. In Geert Webelhuth (ed.) *Government and Binding theory and the Minimalist Program*. Oxford: Blackwell.
- Chomsky, Noam. 1995b. Language and nature. *Mind* 104: 1–61.
- Chomsky, Noam. 1995c. *The minimalist program*. Cambridge, MA.: MIT Press.
- Chomsky, Noam. 2005. Three factors of language design. *Linguistic Inquiry* 36(1): 1–22.
- Chomsky, Noam. 2008. On phases. In Robert Freidin, Carlos P. Otero & Maria Luisa

- Zubizarreta (eds.) *Foundational issues in linguistic theory*. Cambridge, MA.: MIT Press.
- Church, Alonzo. 1936. An unsolvable problem of elementary number theory. *American Journal of Mathematics* 58(2): 345–363.
- Citko, Barbara. 2005. On the nature of merge: External merge, internal merge, and parallel merge. *Linguistic Inquiry* 36(4): 475–496.
- Collins, Chris. 2001. Eliminating labels. In Samuel D. Epstein & T. Daniel Seeley (eds.) *Derivation and explanation in the Minimalist Program*. Oxford: Blackwell.
- Deutsch, David. 1985. Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society of London A* 400: 97–117.
- Deutsch, David. 1997. *The fabric of reality: The science of parallel universes and its implications*. New York: Penguin Books.
- Dornhoff, Larry L. & Franz E. Hohn. 1978. *Applied modern algebra*. New York: Macmillan Publishing Co., Inc.
- Fitch, Tecumseh, Marc D. Hauser & Noam Chomsky. 2005. The evolution of the language faculty: Clarifications and implications. *Cognition* 97: 179–210.
- Formigari, Lia. 2004. *A history of language philosophies*. Amsterdam/Philadelphia: John Benjamins Publishing Company.
- Fortuny, Jordi & Bernat Corominas-Mutra. to appear. Some formal considerations on the generation of hierarchically structured expressions. Ms., Center for Language and Cognition Groningen. Rijksuniversiteit Groningen and ICREA-Complex Systems Lab. Parc de Recerca Biomdica de Barcelona-Universitat Pompeu Fabra; to appear in *Catalan Journal of Linguistics*.
- Hauser, Marc D., Noam Chomsky & Tecumseh Fitch. 2002. The language faculty: What is it, who has it, and how did it evolve? *Science* 298: 1569–1579.
- Hornstein, Norbert. 1992. *As time goes by: Tense and Universal Grammar*. Cambridge, MA.: MIT Press.
- Hornstein, Norbert & Jairo Nunes. 2008. Adjunction, labeling, and bare phrase structure. *Biolinguistics* 2(1): 57–86.
- Idsardi, William J. 2008. Combinatorics for metrical feet. *Biolinguistics* 2(2.2-3): 233–236.
- Jackendoff, Ray. 1977. *X'-syntax*. Cambridge, MA.: MIT Press.
- Jayaseelan, K.A. 2008. Bare phrase structure and specifier-less syntax. *Biolinguistics* 2(1): 87–106.
- Kauffman, Stuart. 1995. *At home in the universe: The search for laws of self-organization and complexity*. Oxford: Oxford University Press.
- Kleene, Stephen. 2002 [1967]. *Mathematical logic*. New York: Dover.
- Korfhage, Robert R. 1974. *Discrete computational structures*. New York: Academic Press.
- Langendoen, Terence & Paul Postal. 1984. *The vastness of natural languages*. Oxford: Basil Blackwell.

- Livio, Mario. 2002. *The Golden Ratio: The story of phi, the world's most astonishing number*. New York: Broadway Books.
- Livio, Mario. 2005. *The equation that couldn't be solved: How mathematic genius discovered the language of symmetry*. New York: Simon and Schuster Paperbacks.
- Lloyd, Seth. 2000. Ultimate physical limits to computation. *Nature* 406: 1047–1054. URL <http://arxiv.org/>. Cornell arXiv; arXiv:quant-ph/9908043v3.
- Lloyd, Seth. 2006. *Programming the universe: A quantum computer scientist takes on the cosmos*. New York: Random House.
- Medeiros, David. 2008. Optimal growth in phrase structure. *Biolinguistics* 2(2-3): 152–195.
- Milne, James S. 2008. Group theory. Lecture Notes, University of Michigan.
- Niyogi, Partha. 2006. *The computational nature of language learning and evolution*. Cambridge, MA.: MIT Press.
- Partee, Barbara, Alice ter Meulen & Robert E. Wall. 1990. *Mathematical methods in linguistics*. Dordrecht: Kluwer Academic Publishers, 2nd edn.
- Pinker, Steven & Ray Jackendoff. 2005. The faculty of language: What's special about it? *Cognition* 95(2): 201–236.
- Popper, Karl R. 1959. *The logic of scientific discovery*. New York: Basic Books, 2nd edn.
- Post, Emil L. 1936. Finite combinatory processes-formulation 1. *Journal of Symbolic Logic* 1: 103–105.
- Post, Emil L. 1944. Recursively enumerable sets of positive integers and their decision problem. *Bulletin of the American Mathematical Society* 50: 284–316.
- Prince, Alan & Paul Smolensky. 1993. Optimality theory: Constraint interaction in generative grammar. Tech. rep., Rutgers University Center for Cognitive Science, New Brunswick, NJ.
- Quine, Willard. 1953. *From a logical point of view*. Cambridge, MA.: Harvard University Press.
- Reichenbach, Hans. 1947. *Elements of symbolic logic*. New York: MacMillan.
- Rosen, Joe. 1995. *Symmetry in science: An introduction to the general theory*. New York: Springer-Verlag.
- Rubin, Edward J. 2002. The structure of modifiers. Ms., University of Utah.
- Rubin, Edward J. 2003. Determining pair-merge. *Linguistic Inquiry* 34: 660–668.
- Soare, Robert I. 1996. Computability and recursion. *Bulletin of Symbolic Logic* 2: 284–321.
- Soare, Robert I. 2007. Computability and incomputability: Computation and logic in the real world. In S. B. Cooper, B. Lowe & Andrea Sorbi (eds.) *Proceedings of the Third Conference on Computability in Europe, CiE 2007, Siena, Italy: Lecture notes in computer science No. 4497*. Heidelberg: Springer-Verlag.
- Soare, Robert I. 2008. Incomputability, turing functionals, and open computing. Ms., Department of Mathematics, University of Chicago.

- Soschen, Alona. 2008. On the nature of syntax. *Biolinguistics* 2(2-3): 196–224.
- Stewart, Ian. 2007. *Why beauty is truth: a history of symmetry*. New York: Basic Books.
- Tarski, Alfred. 1956. *Logic, semantics, metamathematics*. Oxford: Oxford University Press.
- Tesar, Bruce & Paul Smolensky. 2000. *Learnability in Optimality Theory*. Cambridge, MA.: MIT Press.
- Tomalin, Marcus. 2006. *Formal foundations of generative grammar*. Cambridge: Cambridge University Press.
- Turing, Alan M. 1936. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society* 42: 230–265.
- Uriagereka, Juan. 1998. *Rhyme and reason: An introduction to minimalist syntax*. Cambridge, MA.: MIT Press.