

April 3, 2009

1 Human Language and Mathematical Systems

Language is not math and languages are not numbers—this is intuitive enough. It may not be so intuitive to say that natural human language cannot be exhaustively modeled by a formal mathematical system. What do I mean by “modeled exhaustively?” The answer to this lies in how we define the crucial properties of human language and what our scientific goals are.

If I want to measure how many times the word “bedeviled” occurs in the modern English usage, then there are statistical techniques designed to calculate and measure such patterns based on a certain corpus or databank. This kind of study deals with the external distribution patterns of words and sentences and is largely dependent on cultural and historical accidents. For example, the word ‘rad’ may have had a larger, or more frequent, distribution in the 1980’s than it does now. It is simply a historical and cultural “accident” that ‘rad’ has the overall distribution that it does. The fields that study these kinds of matters are named lexicography and corpus linguistics, and the simplified example given here does not do justice their full complexity and richness, but it works to illustrate my point that the external output of human languages are subject to various kinds of measuring techniques with varying degrees of mathematical precision in showing language patterns. Importantly, these patterns (for example, how many times the word ‘rad’ is used) are not due to properties of Language itself, but to other sociological and historical-cultural properties. In other words, how many times a word is used in a language is not a crucial property of either that language itself, or of Language in general. Lastly, Frequency is not all that can be studied in lexicography and corpus linguistics, and in fact, some characteristics of human language can be revealed through these approaches (such as the size of a vocabulary). This still does not change the fact that, while these kinds of studies are important and can be quite complicated, they are not useful for those who wish to discover exactly what Language is and why it is a unique property of humans.

There is another mathematical way to study human language that might be relevant to explaining the *what* and *textslwhy* of human language. The study of what are called logical or formal languages has been used to explain human language. Examples include computer programming languages such as C++, or what is called the first order predicate calculus. In these languages efficiency is the golden rule. One can consciously design them to filter out redundancies and

ambiguities. Focusing on the logical system of a first order predicate calculus, we might try to package natural human languages into this artificial logical language. Or, we might try to *reduce* natural language to the sleek form of the logical system. This is what is meant by ‘exhaustively model’ natural language by a formal system, and it cannot be done as I will show.

1.1 Ambiguity and Redundancy

Formal or logical languages are very different from human languages in at least two major ways: redundancy and ambiguity.

- (1) Jon runs and we like to run too.
- (2)
 - a. Jon paints cars.
 - b. Visiting relatives can be lame.
 - c. new drug technology

In (1) there are two kinds of redundancy. The first is in the redundancy of the sound of the distinct words ‘to’ and ‘too.’ In speaking, this kind of redundancy can actually lead to ambiguity—think also of the word ‘two.’ The second kind of redundancy in (1) is a more complicated kind known as agreement. In English, we have subject-verb agreement where the the same kind of information shows up on both the verb and the the subject. This information includes the number (singular or plural), gender (masculine, feminine, and neutral), and person (first, second, third). In the example given, ‘Jon’ is a singular, masculine, third person noun and so the verb that describes Jon’s action must also have singular, masculine, third person features. For this reason, we know that it must be the verb ‘runs’ that describes Jon’s action and ‘run’ that describes the action of ‘we.’ In languages other than English the agreement system can be much more complex—coding dual and various kinds of plural numbers, as well as a wide variety of different kinds of grammatical gender such as long, thin, animate, and also different kinds of person such as first-inclusive and first-exclusive. No matter what kinds of features or coded by agreement in natural languages, it seems pretty clear that natural language have to have agreement. Differences arise in the form of agreement, with some systems being observable and others not. Despite these differences, agreement in languages is a form of redundancy not found in formal languages—it is not efficient in an operational way in mathematical, logical, or computational languages. But in natural human language, there are strong arguments based on pretty good evidence that the redundancy of agreement represents an optimal so-

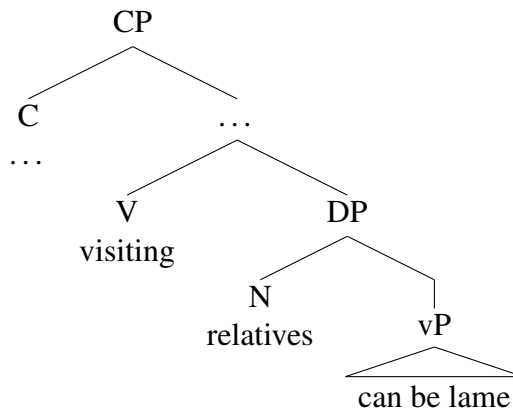


Figure 1: (Visiting (relatives (can be lame)))

lution to the different systems needed to run the language generator. Example (2) shows ambiguity. There are two types of general ambiguity, one is lexical and one is structural. In example (1) the words ‘to’ and ‘too’ sound alike but have different meanings—also ‘two.’ This is an example of lexical ambiguity. Think of other homonyms—words that sound alike but have different meanings—like the animal ‘bat,’ the baseball ‘bat,’ and the verb to ‘bat’ your eyelashes. Structural ambiguity is shown in the examples in (2). In (2-a) does Jon apply paint to cars or does he apply paint to a canvas while drawing cars? In (2-b) is it lame to go visit relatives or is it lame when visiting relatives come stay with you. Finally, in (2-c) are we talking about technology for new drugs, or new technology for old drugs? The answer to these questions can go either way, depending on your interpretation of the ambiguity. It is commonly understood in theoretical syntax that ambiguity provides robust support for the notion that there is an abstract hierarchical structure underlying natural language syntax. Simplified examples can be seen in the syntactic trees of Figures 1 through 4—category labels are not meant to be exact, I’m only trying to show different hierarchical groupings. Such tree structures are geometric models for the hierarchical relationships between constituents. There are no linguistic claims that such trees exist “in the head,” but the hierarchical relations between constituents that they model *are* assumed to be real properties of the the human mind.

Formal languages simply do not have the properties of redundancy and ambiguity because they make systems less efficient. We can build a logic or computer language with ambiguity in it but it might not be very useful. What linguists have found useful is to use tools from logical systems to model parts of natural lan-

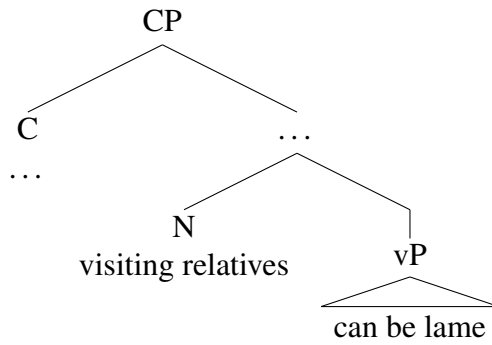


Figure 2: (Visiting relatives (can be lame))

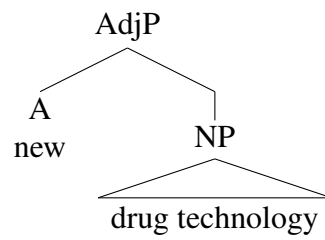


Figure 3: (new (drug technology))

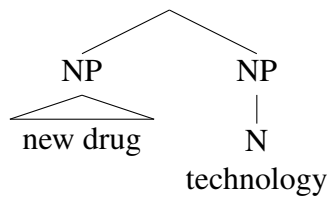


Figure 4: (new drug (technology))

guage. This appears to work very well. When it comes to modeling all of natural language by a formal system, things quickly get out of hand. In fact, in computational linguistics—the field of linguistics that applies linguistic discoveries to computer languages—one of the biggest problems is ambiguity resolution, both structural and lexical.

1.2 Competence-Performance, Internal-External

Around the 1930's attempts to model natural language by an efficient and consistent model of grammar quickly showed that the model 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—think of the example of the word of 'rad. Neither (a) nor (b) gave answers to the *what* and *why* of natural language. In other words, they did not shed any explanatory light on the following observations: any human being of sufficiently young age can acquire any human language with only a few years of exposure. Statistical studies have shown that the amount of language data a normal child is exposed to is not enough for the child to learn that language by a simple method. Some had thought that children begin as blank slates simply mimic their environment, or that they learn certain basics of a language and then extend on the base. But the fact is that the amount and quality of language data the child is exposed to are not significant enough to allow a child to start from a blank slate and learn everything the need by the age of about five. Not only this, but a child can speak sentences they never heard before. Furthermore, a child can understand a great deal more than they can speak. This helped support a growing sense of the distinction between language *performance* and language *competence*. The latter is the grammatical system that we all seem to have intuitively, and the former is how this grammatical system is used in the world. Here we will take a short detour in order to see an example showing the difference between these two distinctions.

A modern speaker of English will rarely employ the infamous 'who/whom' distinction. Most people do not remember how the 'who/whom' distinction is to be used and will avoid it whenever possible. On the other hand, nobody has to think about the 'he/him' distinction. We all know that the sentences

- (3) He is the soccer player.
- (4) He did not intend to kick him.

are perfectly fine, but that

- (5) *Him is the soccer player.
- (6) *Him did not intend to kick he.

are ill formed sentences. The ‘who/whom’ distinction codes exactly the same information: he-who, him-whom; notice the *-m* ending. So how do we account for the massive difference in performance here? How do we account for the fact that practically no one can remember grade school teachers and grammarians scolding us on the proper use of ‘he-him,’ but most everyone I know can certainly remember having to pay extra attention the ‘who/whom’ distinction? The answer is quite simple, really. The ‘who/whom’ distinction is not part of our competence for English anymore, while the ‘he/him’ distinction is. I will explain. The English language has been losing what is called its Case system. Case is a way for languages to, essentially, tell us who did what to whom. Case information is almost always coded on nouns and helps distinguish, among other things, subject and object. In current syntactic theory Case plays a very important role in forming all sentences, and all languages must have some form of Case even if it does not show up on the nouns—it is usually called structural case and is distinguished from the case properties that we can see, called morphological case. In English, we used to have a very rich morphological case system but have been losing it for reasons too complicated to explain here (but see Appendix xxxx). However, the case system in English pronouns is still preserved: he, she, him, her. The different forms of the pronouns signal to us whether or not they are subject or object, no matter where they may occur in the sentence. For some reason this preservation of the case system does not seem to apply to pronouns that do not specify grammatical gender: it, who/whom. For this reason, modern English speakers do not intuitively code the ‘who/whom’ distinction because it no longer belongs to the actual grammatical system of English—or what used to be called our language competence. This is why we all have to struggle to remember, or memorize, that ‘who’ is for subjects and ‘whom’ for objects. Some grammarians decided that the ‘who/whom’ distinction was still important and so they tried to tell everyone that the distinction was important. Clearly, telling people how to speak does not work if what you are telling them goes against their native intuitions; and we all have native intuitions about the English language, despite what prescriptive grammarians may try to tell us about the right and wrong way to speak. The grammarian rightness or wrongness is a matter of socio-political and cultural standards, NOT an objective standard of the nature of the English language—or any language for

that matter. The competence-performance distinction is very difficult to define and it is probably not as cut-and-dry as it appears to be. The terms are not usually used in current syntactic theory, at least not in a really technical way, mostly due to the problems in trying to specify what parts of language belong to competence and what parts to performance. I have always been more inclined to see it as a duality than as a complementarity that can only be elaborated after much more is known about Language—how it develops in children from birth, how it changes over generations, what kinds of social factors can influence sentence structure. In other words, I see Language as partly defined by a competence-performance duality such that the grammatical system of a language may be altered by speakers’ performance/use of that system. Another way to describe this may be the statement that Language is a system composed of some finite invariant principles and some finitely variable parameters that have a small range of variability. The competence-performance duality belongs to the parts of language that are parametric.

Returning now to the problem of mathematical models in language. It soon became obvious (by about the early 1960s) that no simple mathematical model—whether it was a statistical distribution or a logical calculus—could capture the structure of Language. Eventually, conceptual tensions between (a)—synthetic grammar of a first order predicate calculus and (b)—distribution of morphosyntactic elements in language data, as well as between the notions of *competence* and *performance* forced a distinction between Internal and External structure known as I-language and E-language (around early 1990s). The focus of most generative syntacticians is in describing and explaining the internal mechanisms responsible for initial language acquisition. This is the **what** and **why** of Language and belongs to the domain of I-language. In the current climate of research it is now more appropriate to talk of the “growth form” or development of language in terms of acquisition. Humans do not really acquire language, unless we want to talk about humans acquiring vision, hearing, walking, and other kinds of processes. These processes clearly must develop in response to the environment and need guidance from the adult in order to reach maximum potential, but they can also develop on their own. The same is true of language—it is just that the *apparent* variety of particular development paths (that is, languages such as Hindi, Swahili, or Hopi) seems stunningly distinctive when compared to the variable paths that vision, hearing, or walking can develop toward. What makes studying Language more challenging than vision or hearing is that, for one, the results of development seem to be so drastically different (in the form of different languages), and two, there is no easily detected material organ to which we can attribute the cause, and

thus, put under a microscope like we do with eyeballs for vision, ear canals with hearing, and general physiology of legs, hip, and balance for walking. It should also be pointed out that the processes responsible for vision, hearing, and walking are only basically understood—much is still unknown.

I-language, as a growth form unique to the human organism, is a natural object—or so the argument goes. What makes this conjecture so contentious, it seems, is that unlike vision, hearing, or walking, it appears to show not only a great variety of resulting development paths, but also it seems to be unique in the animal kingdom. However, some of this contention can be resolved by noticing that Language is employed as a communication system and the animal kingdom has an extremely wide variety of communication systems. Within this variety, we might also assume that at least some of these communication systems did not evolve as an adaptation for communication, but may have been co-opted as such. In this context, the uniqueness of human language starts to look more like a system unique only because human biology is unique within the animal kingdom. That is, the unique status of Language is dependent on the unique status of human biology—and in particular, neurobiology. As such, I-language as a natural object, just like any other object of nature, can be modeled by various formal or mathematical systems with varying degrees of success and relative to the specific goals of the research. However, I emphasize the point 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. Consequently, we should not expect any formal/mathematical system to exhaustively model natural language; but useful methods for mathematicization should always motivate us.

1.3 Mathematicization

In the rest of this chapter I will actually show how linguists go about deciding whether or not some mathematical tool or theory is relevant to natural language. However, I should point out now, as I will continue to do, that linguistics is its own field of study and has as its object of study a particularly unique object in all of nature. Arguments for the adoption of certain mathematical tools are not arguments for the wholesale adoption of the mathematical enterprise to which those tools belong. The assumption here is that because of the uniqueness of human language AND because attempts to exhaustively reduce human language to known formal systems has failed, linguists are going to have to create their own kind of formality. To a large degree, this has already happened, but the formal

system is by no means adequate and it needs to be mathematicized to whatever degree is possible. In the history of science this is not that strange—new mathematics have either been developed or resurrected to meet the demands of a new science. For example, Claude Shannon and others took George Boole’s binary algebraic logic—which had essentially been un-influential in the world of logic because Frege was able to construct a much more useful logic based on the use of quantifiers such as ‘All’ and ‘Some’—and applied to electronic circuits to get the rudimentary base of all computers. Another example includes Einstein’s use and elaboration on non-Euclidean geometries—which had been seen as an interesting but impractical geometry—for the exposition of his general theory of relativity. A last example is Benoit Mandelbrot’s fractal geometry—which was first viewed as a somewhat interesting, if not superficial, geometry that made cool looking computer images but had no real applications. Needless to say, applications of fractal geometry have already found wide acceptance from their use in shaping the antennae of cell phones and blackberries to the measurement of numerous natural phenomena such as lightening, trees and forests, coastlines, and even heartbeat rhythms. What such a new mathematics for linguistics would look like is not certain, if possible at all. But the reasoning that states “If Language is a natural object then it should be able to be modeled by a mathematicized system” is not very controversial. In fact, this could be stated as a corollary to what I have termed the Weak Minimalist Thesis in § refsmt:sec. What *is* controversial is the status of natural language as a natural object—but for the purposes of this book we have already agreed to assume that it is a natural object (for dissenting opinions see the general literature from cognitive or functionalist linguistics). In this way, then, it is not so much that Language cannot be modeled by a mathematicized system, but that a mathematicized system does not yet exist that could model Language. Also, notice that when I am speaking about a formal model for Language I use the term ‘mathematicized’ and not ‘mathematical.’ This implies that whatever rigorous system scientifically minded linguists find useful may not actually be a mathematical one by nature. If linguistic models seek to be appropriately scientific they need to be able to be translated into a mathematical system, that is, mathematicized. The issue of finding an ‘exhaustive’ model for language is probably more an issue of science in general—Is any scientific model exhaustive of the phenomena it represents? I hazard a guess that no scientific model can live up to these standards, but it is nice to try. By these standards, then, an exhaustive model of something so intricately connected to human consciousness and cognition—not to mention things like free will—could probably never even be exhaustively **described**. But this is not the linguists job. The job of the linguist is to give the **what** and **why**

of Language—to explain the abstract principles and parameters of the language faculty (the human ability to develop and use particular languages). The really interesting areas of languages, such as poetry and literature, are out of bounds. A strictly exhaustive model would have to include this out of bounds area and if we figure that a description covers less empirical ground than an explanation, then we have reason to believe that any goal of a strictly exhaustive model of human language is highly unlikely. On the other hand, an exhaustive model of parts of human language seems more reasonable—especially if we limit these parts to abstract principles and parameters of the human language faculty.

As it stands now, with recent formulations of syntactic theory and interest in notions of optimal form in nature—such as the Fibonacci pattern and types of symmetry—the conscious effort to view Language as a natural object has started seeing some immediate and interesting results. However, it needs to be stressed that all results are, or should be, viewed with a very critical eye. Many of the arguments are very abstract and so is the evidence used to justify such arguments. Nonetheless, if one is committed to a research program like the Minimalist Program and an idealistic approach to science like the Galilean approach, then the results one gets are suggestive of future progress. This progress includes, in fact demands, that one day we will find a way to empirically test all of our hypotheses. The rest of what follows in this chapter is based on ? and represents results along the lines just discussed. They are speculative, abstract, and do model any actual natural language data. Instead, they investigate the possible mathematical structure of the operation Merge in syntax. For those of you who do not like math feel free to skip the rest of the chapter. Another reason for skipping the rest of this chapter is that you do not know anything about theoretical syntax. While lack of knowledge should never hold someone back from trying to gain knowledge, sometimes it can be a frustrating experience when the lack of appropriate educational background and training gets in the way.

2 On With the Formalities

Intense research into Fibonacci (F) patterns arising from ideal iterations in both prosody and phrase structures has shown that there is a special role for rigorously modeling the computability requirements of producing such patterns. F patterns produce self-similar structures along repeated and computable iterations. The computable iterations of the F sequence can in turn produce idealized phrase structures, as ? and ? have shown. These idealized phrase structures also map pro-

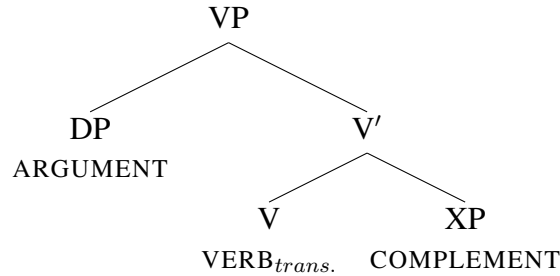


Figure 5: Simple argument structure

jections 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 phrase 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 § 3.2. In the context of these developments it is only natural to look towards the formal study of symmetry for possible rigor.

It is well-understood that 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 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 ? and ? discuss. If we iterate this process, unrestricted, then the projection¹ 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.² 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; see Fig-

¹This is not the projection of mathematics.

²Of 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 $\{a, b\}$; see ?, ? ?, and ?? for examples.

ures 1 to 6. The reason for focusing on symmetry, then, is to have some criteria by which to measure symmetry-breaking.

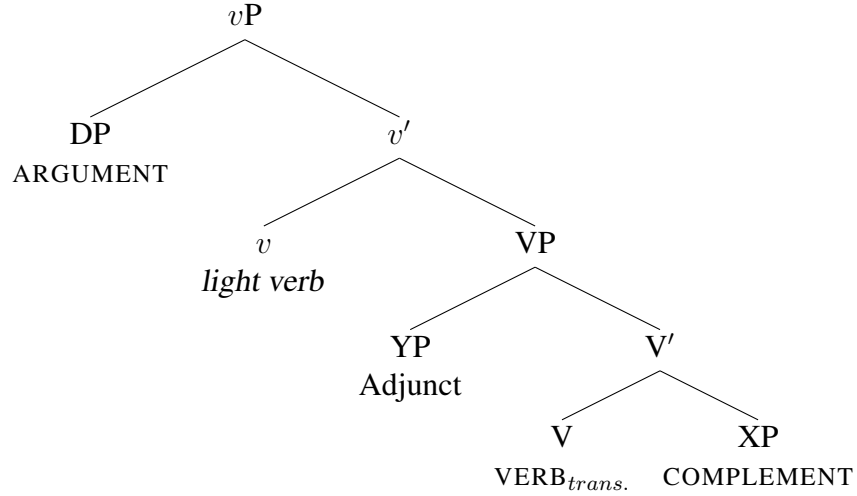


Figure 6: Complex argument structure with v P shell

3 α Merge and symmetry

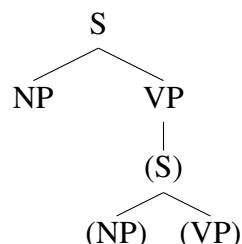
3.1 Ambiguous Merge

Given the naturalistic and rigorous framework for linguistic study briefly discussed in § ??, it is reasonable to experiment with the use of different kinds of rigor to explain Language. The search and justification for rigor in generative linguistics has a key issue since the beginning. Tomalin has an excellent introduction to the formal foundations of generative theories in his book *Formal Foundations of Generative Grammar*. In it he relates the following episode.

In 1954 Yehoshua Bar-Hillel published a paper, ? in the journal *Language* about the importance to linguistics of the then current practices of logical syntax and logical semantics. This paper was published within a growing context of linguistic interest in formal/logical methods applied to natural language data. A short while later the up-and-coming linguist philosopher Noam Chomsky published a response to Bar-Hillel in the same journal, ?. In this article Chomsky expressed doubts about the ability of formal systems to model language behavior.

He particularly expressed doubt about the usefulness of certain kinds of *recursive definitions* that Bar-Hillel mentioned, particularly in reference to ?.

(7) $S \rightarrow NP VP (S)$



(8) a. $NP \rightarrow N (Det) (NP) (\mathbf{PP})$
b. $PP \rightarrow P (\mathbf{NP})$

However, only a few years later Chomsky would make use of recursive definitions for defining hierarchical relationships of constituents. Examples can be found in (7) and (8).³ Such informal definitions by “recursion” proved to be problematic for describing the general environments of unique linguistic constituents—to the effect that they were understood to be too strong. Unique constituent definitions were quickly generalized under the X-bar schema first proposed in ? and later extended by ?. Years later, the X-bar model was modified and streamlined by ? and ? 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 ?, ?, ?, and ? to name a few.

(9) X-bar generalization for constituents
a. $X'' \rightarrow \text{Specifier}, X'$
b. $X' \rightarrow X, (\text{Complement})$

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.

³? would come to recognize, specifically 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).

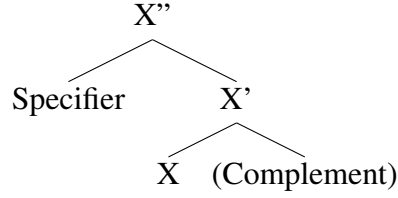


Figure 7: First X-bar structure

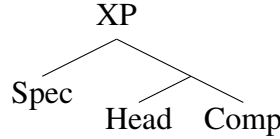


Figure 8: General BPS X-bar model

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 ?, and I will denote its operation as αMerge .

Definition 1. $\{\{\alpha\}, \{\beta\}\} \xrightarrow{\text{merge}} \{\Lambda, \{\alpha, \beta\}\} = K = \alpha\text{Merge}$ ⁴

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 ? and ? 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.⁵ In tree form, αMerge produces a mirror symmetric object, seen in Figure 9.

⁴Where 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 $\{\alpha, \beta\} \in K$, but only $\{\{\alpha, \beta\}, \Lambda\} \subseteq K$; see ?.

⁵It should be noted that αMerge , as presented here, may be operable in non-human cognition. See ? for more on this, as well as arguments against ?. Also note, αMerge as presented here does not equate directly to adjunction defined in ?, Chapter 3, as a time-dependent insertion $\{\{\alpha\}_t$,

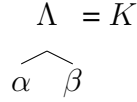


Figure 9: α Merge tree produces object K

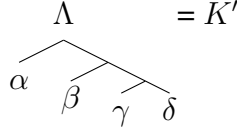


Figure 10: Iterated α Merge tree produces object K'

With mirror-symmetric objects each side is the same in terms of their relative position and relation to the other side. Figure 9 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 10—notice that K' is a compound of two mirror-symmetric trees but itself has no bilateral symmetry.⁶ Once the bilaterally symmetric object of Figure 9 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 10, $\{\alpha, \beta\}$ 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 accu-

$\{\beta\}_{t+1}$ with an inherent temporal asymmetry. Although, there doesn't 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 § 4.3 for more detail.

⁶In fact, the tree in Figure 10 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 ? of the relation between *string length* of n and *depth* of $(n-1)$.

mulated as the result of iterated Merge operations correlates with the Fibonacci pattern.

3.2 Fib(n) levels and the X-ratio

3.2.1 Rationale and justification

The idealized objects produced in Figures 9 and 10 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 ? and ? have shown; see also ?, ?, ?, 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, ? 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, ? is right, F patterns are undeniable and their investigation in human syntax is significant—if approached soberly.

3.2.2 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 ?? and Table ??—the latter of which generates the tree in Figure 14; see ? 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 ? and ? 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 ?? 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.⁷ 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 doesn't matter and can be changed.

The GS applied to phrase structure embedding gives similar results to ?, 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 ?? 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 13.

By bracketing the phrase string produced from Fibonacci level 34 (F_{34}) differently than it is parsed in Table ??, a fully balanced binary tree, like the one in ? and ? can be derived. Figure 11 is an example of a Medeiros tree and Figure 12

⁷The 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...

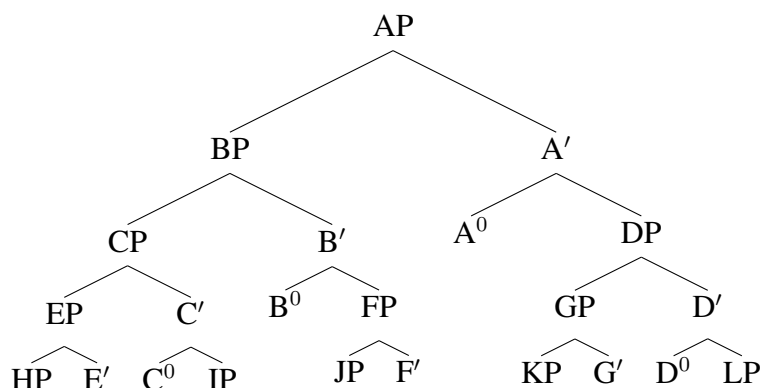


Figure 11: Medeiros tree

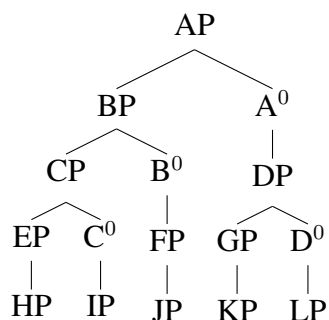


Figure 12: Medeiros tree with no Spec

shows the same tree without specifiers. The Phrase Strings 11 and 12 represent the latter trees, respectively; 11 for Figure 11 and 12 for Figure 12. 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 13. 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.⁸

In Figure 14 I produce a binary right-branching alternative to Medeiros' balanced binary tree.⁹ This right-branching alternative is based on a well-known property of F patterns called the Golden Sequence; for example ?, but especially

⁸Potential 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 13; compare the right-branching tree of Figure 14 derived from a straightforward sequential embedding of F^{34} .

⁹Also work by Alana Soschen, for example ?.

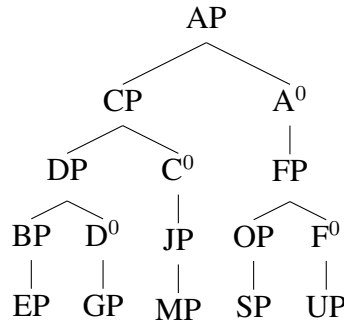


Figure 13: Randomly projecting α Merge tree adapted to Medeiros string

?. It has the same F pattern for syntactic objects that derivations in ? 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 14 as the Medeiros equivalent in Figure 13. 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 (10), but for terminal heads the sequence begins as shown in (11) (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.

(10) Fib sequence for XP: [1, 1, 2, 3, 5, ...]

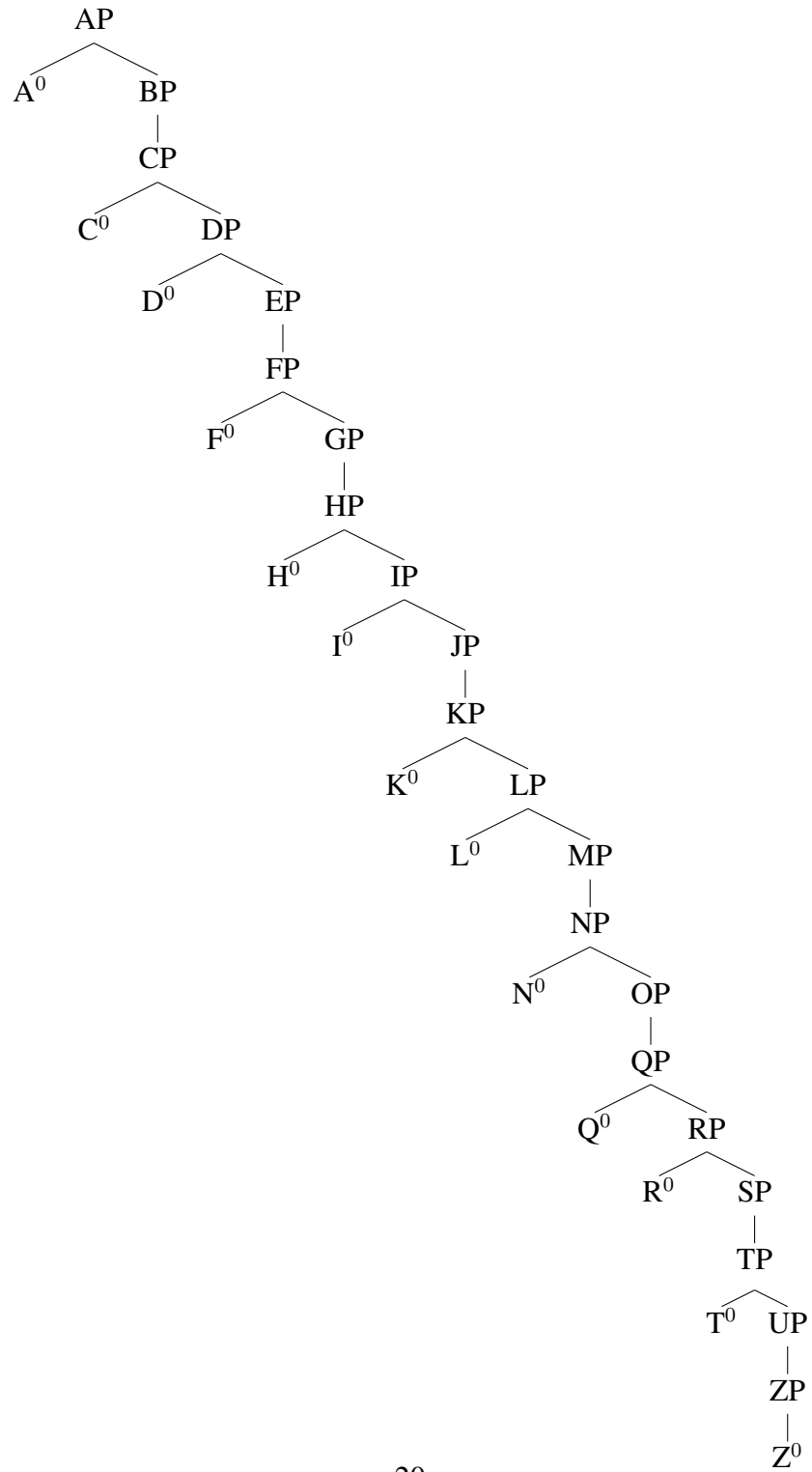


Figure 14: Random label projection of α Merge

(11) 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 F^3 string in Table ?? has one X^0 and two XPs (A^0 , α , and β) such that $2/1 = 2$. Looking at F^34 , 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 ?, ?, ?, and ?. 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 $\text{Fib}(n : n-2)$.

3.2.3 Symmetry in syntax

Group Theory is the formal application of study for the investigation of symmetry in nature; see ? and ? for popularized accounts and ? and ? for formal introductions—also Chapter 10 in ?. 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 ?.

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 $Fib(n \pm 2)$, which determines the X-bar level at which the F pattern begins, as in (10) and (11). This is generally consistent with the empirical and theoretical claims of the hyper-minimalist BPS of, for example, ?, 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 § 3.1. 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 § 3.1 do share one thing in common: they are both symmetries of syntactic objects and not of, say, syntactic rules of the sort discussed in ?.¹⁰ Thus, heads and phrases are symmetrical in an important way that differs from traditional notions of symmetry in syntax.¹¹ 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

¹⁰For 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, ? highlights how this observation can be traced back to the *Structure Preservation Hypothesis* of Emonds. This seems like a significant advance in syntactic theory.

¹¹Such as subject-object asymmetries.

law for efficient self-organization.¹² These laws or first principles may be coupled with a combination of interface constraints and/or syntactic operations, such as Agree in ?, that essentially work to break the initial symmetry of ambiguously merged elements or elements that have the potential for head-phrase (X-ratio) ambiguity.¹³

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.

3.2.4 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. ? has shown how a rotational symmetry can be used to model symmetrical relations between projections and chains (pg. 46), see also ?, 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 ? for discussion of fractals in nature and the Minimalist Program. ?, ?, ?, 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 ($= 1.618033\dots$).¹⁴ This value is also the ratio responsible for making symmetrical spirals and other types of symmetrical objects. If one takes seriously the notion

¹²This kind of thinking meshes very well with ? 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.

¹³For example, clitics or nominal predicates.

¹⁴When 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.

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.

4 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 ?. 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 leui 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 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.

4.1 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 ? and ? 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 ?, ?, ?, ?, and ?. 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 ?, ?, ?, ?, ?, ?, ?, ?, and of particular relevance here ?. 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 ? 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).

? expressed a similar attitude in response to arguments by ? for the use of formal models in natural language—specifically the use of recursive definitions in a formal syntax of the type found in ?:¹⁵

[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*

¹⁵Interestingly, as ? points out, Bar-Hillel's linguistic work served as the foundation for a specific approach to natural language, Combinatorial Categorical Grammar (CCG), used in computational linguistics; for example ?.

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 (pg. 45; 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 ?, ?, ?, ?, and also ?, 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 ?.¹⁶

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 ?, ???, ?, ? and many others past and present; see also ? 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.

¹⁶The 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.

4.1.1 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 ?. However, ? 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.

? 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 (pg. 100)

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)finately to produce various forms (i.e., discrete infinity). For example, ? 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.

4.2 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.”¹⁷?, and more recently, ??, have pointed out a historical

¹⁷One example of “recursive” can be found in ?, who show that a recursive definition can be gotten by considering

the set M for all mirror-image string 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 (pp. 181-82).

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. Q_0

confusion in the use of the two terms that may lead to unfortunate misunderstandings relative to the field of study and the definitions 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. ? 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’” (pg. 34). 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 ? and ?, although ?—? 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 (pg. 25).

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.

Recall that ? pg. 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 ?, also in ?: “every *generated set* is effectively enumerable, every effectively enumerable set of positive integers is recursively enumerable” (pg. 286; 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, ?, and ?.¹⁸ 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, ? 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

¹⁸? state for the bracket [_AB 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).

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 (pg. 29).

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 ? 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 succesful in theoretical syntax. An anology can be made to the linguistic use of Turing machines (TM): theoretical syntacticians are more concerned with a specific type of universal TM¹⁹ as it relates to the operation Merge within the larger backdrop of the concern for algorithms that are both physically tractable and computably efficient.

4.3 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

¹⁹This is the Turing *automatic*-Machine, compared to a stochastic/probabilistic TM, or even a quantum TM that ? formalized.

collection of sets, can be a group if it can be defined by Definition 4; the following is loosely based on ?.²⁰

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 10 in ?; also ?, section 39 in ?; also ?, ?—or any abstract or modern algebra book; also ? and ? 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 .

²⁰Notice and interesting parallel with the criterion given below and some examples from ? 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.

I now briefly sketch how αMerge 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 αMerge 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 αMerge 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 αMerge —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, $?$. 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 αMerge), 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 10, 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.

5 Final Remarks

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 (9). 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, ? 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 ?, ?, ?, ?, ?, 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 ? and ? 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 ? 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 ?. Following this, I focus on arguments made by the mathematician and computability theorist ??? 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.