Evolutionary Psychology

www.epjournal.net – 2013. 11(3): 470-492

Original Article

A Framework for the Comparative Study of Language

Juan Uriagereka, Department of Linguistics, University of Maryland, College Park, Maryland, USA. Email: juan@umd.edu (Corresponding author).

James A. Reggia, Department of Computer Science, University of Maryland, College Park, Maryland, USA.

Gerald S. Wilkinson, Department of Biology, University of Maryland, College Park, Maryland, USA.

Abstract: Comparative studies of language are difficult because few language precursors are recognized. In this paper we propose a framework for designing experiments that test for structural and semantic patterns indicative of simple or complex grammars as originally described by Chomsky. We argue that a key issue is whether animals can recognize full recursion, which is the hallmark of context-free grammar. We discuss limitations of recent experiments that have attempted to address this issue, and point out that experiments aimed at detecting patterns that follow a Fibonacci series have advantages over other artificial context-free grammars. We also argue that experiments using complex sequences of behaviors could, in principle, provide evidence for fully recursive thought. Some of these ideas could also be approached using artificial life simulations, which have the potential to reveal the types of evolutionary transitions that could occur over time. Because the framework we propose has specific memory and computational requirements, future experiments could target candidate genes with the goal of revealing the genetic underpinnings of complex cognition.

Keywords: Chomsky Hierarchy, context-free grammar, syntax, semantics, recursion

Introduction

The existence of language poses difficult questions for comparative psychology. This is both because non-human species lack hallmarks of human language and also because researchers across disciplines often have different ideas of what language ultimately is. Comparative studies would be facilitated if a framework existed in which different forms of behavior could be logically related in a formal manner. Such a framework could then be used to design experiments aimed at testing hypotheses about the evolution of language precursors.

Our approach, couched within the Computational Theory of Mind, is based on a formal distinction among "levels of complexity" in syntactic structures: the so-called Chomsky Hierarchy. Originally proposed for classifying formal languages according to the computational power it takes to generate them, this approach provides a framework in which structured sequences can be described via their computational complexity. From this perspective, a key comparative question is whether other animal behaviors can be classified at different levels within the hierarchy, thereby presupposing their computational nature.

This question turns out to be hard to answer. The present approach was catalyzed by reactions to a report (Gentner, Fenn, Margoliash, and Nusbaum, 2006) that European starlings can master a level of computational complexity that enables recursion, a key step for linguistic creativity (Hauser, Chomsky, and Fitch, 2002). Our paper critically examines this result and others, and suggests alternative approaches to experimentally answering the question just posed.

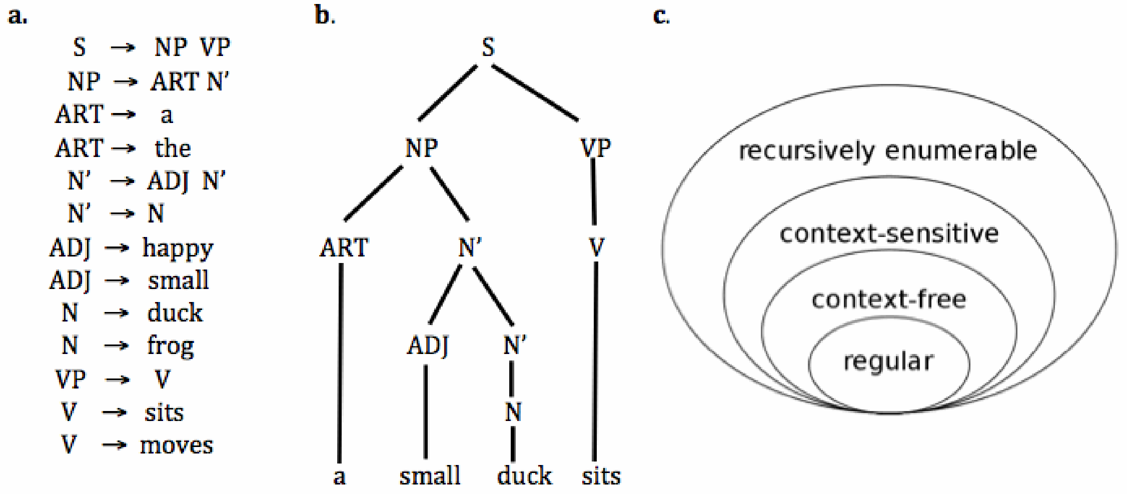
The Chomsky Hierarchy

One approach to the study of (human) language is through the study of *formal languages* and corresponding grammars (Jurafsky and Martin, 2000). A *formal grammar* is a procedure that operates on a finite set of symbols indicating, through an explicit set of rules of production, how to concatenate some of those symbols into a sequence. For example, that very last sentence could be seen as a concatenation of typographical elements like "h," "o," "w," "space," "t," "o," and so on. A formal language need not have any meaning, or for that matter any recognizable symbols; we could designate, for example, each of the pages in this article as a relevant symbol, and their concatenation would produce the sequence of pages readers have in their hands.

A formal grammar *generates* a language: By beginning with a start symbol and repeatedly selecting and applying its rules, it is possible to generate strings/sentences in the language. We call such a sequence of rule applications a *derivation*. By definition, grammars as just described are finite; however under certain circumstances they may generate languages (sets of strings) of infinite size. Figure 1 gives an example of a formal grammar and some examples of strings in the corresponding language it generates.

It is possible to divide languages/grammars into four types, forming the Chomsky Hierarchy (Chomsky, 1956). Table 1 lists these grammar types ordered in terms of increasing complexity. Each type is characterized by restrictions on the form that its rules can have. Letters *X* and *Y* in the Table designate arbitrary single non-terminals (such as NP, V, etc., in Figure 1a), while the letter *a* stands for an arbitrary terminal symbol ("duck," "the," etc.). Greek letters α, β, and γ represent arbitrary strings of symbols composed of both terminals and/or non-terminals.

Figure 1. (a) A small, simple grammar consisting of 13 rules or productions that collectively define or *generate* a specific language; **(b)** illustration of a parse tree generated by this grammar; **(c)** Subset relationships between the four types of languages in the Chomsky hierarchy (see text for details).



Notes: (a) Rules are written in terms of *non-terminals*, symbols that must be rewritten (expressed here in upper case, such as “NP” and “ART”), and *terminals*, symbols that cannot be rewritten and which thus form sentences in the language that one ultimately observes (expressed here in lower case, such as “the” and “sits”). In this specific grammar, each rule has a non-terminal symbol on the left, followed by an arrow that indicates that this symbol can be rewritten using the non-terminal and/or terminal symbol(s) on the right. For example, the first rule indicates that a sentence S in this language consists of a noun phrase NP followed by a verb phrase VP. Other non-terminals are for articles ART, parts of noun phrases N’, adjectives ADJ, nouns N and verbs V, while *terminals* here are lower case English words like “the” and “duck.” The set of strings/sentences this grammar can generate is its language. For example, this language includes “a small duck sits” because that sentence can be generated by starting with S and using eight of the listed rules to generate a parse tree as illustrated in (b). This specific language is infinite due to the recursive rule N’ → ADJ N’ which defines an N’ in terms of itself. Thus the language includes sentences with an arbitrary number of (repetitive) adjectives, such as “the happy happy small frog moves”

Table 1. The Chomsky Hierarchy of Languages

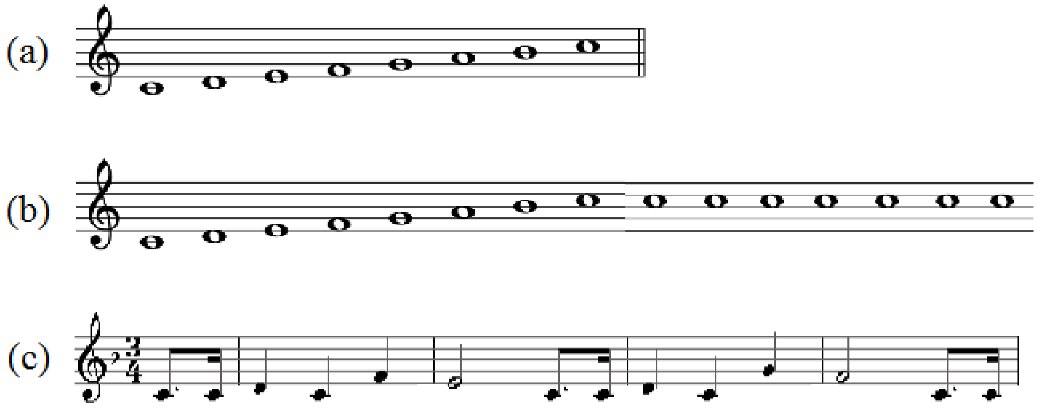
Type of Grammar	Form of Rules	Corresponding machines
Regular	$X \rightarrow a,\ X \rightarrow a\ Y$	Finite state machines
Context-free	$X \rightarrow \gamma$	Pushdown stack automata
Context-sensitive	$\alpha\ X\ \beta \rightarrow \alpha\ \gamma\ \beta$	Linear bounded automata
Recursively enumerable	$\alpha \rightarrow \beta$	Turing machines

Regular grammars are highly restricted to contain only rules in which any non-terminal symbol *X* is replaced by either just a terminal *a*, or by a terminal *a* followed by an additional non-terminal *Y*. The grammar in Figure 1, for example, is *not* a regular grammar, because it contains rules such as S → NP VP where the non-terminal S is rewritten as two non-terminals, which is not allowed in regular grammars. Regular grammars are relatively simple in that the range of languages they can generate is in a sense quite limited.

Context-free grammars are less restricted in terms of the languages they can generate. They contain rules in which any non-terminal symbol X can be rewritten as an arbitrary string of terminals and non-terminals γ. For example, the grammar in Figure 1a is context-free. The term "context-free" indicates that replacing X with whatever γ appears on the right side of a rule can be done regardless of the context in which X occurs.

Context-free languages are more complex in terms of their structure and the computations needed to produce them than regular languages. Intuitively, regular languages are either very fixed in their form (e.g., the string of characters in Figure 2a) or if they have variations in form, this is limited to monotonous iteration (as in Figure 2b). In contrast, in a context-free language there could be entirely open-ended forms of variation (see Figure 2c).

Figure 2. (a) The sequence of notes in the familiar tonal scale are what they are, in this particular order and without any possible variation; **(b)** Any given note could be repeated endlessly through some sort of loop, such as the last note at the end of a scale; **(c)** However, the simplest tunes are not expressible in those fixed terms: even the humble Happy Birthday has a phrasal structure that contains slight variations, and further variations could of course be composed to continue the tune.



We will focus on regular and context-free grammars, although there are two other types of grammars that generate even more complex languages (*context sensitive grammars* and *recursively enumerable grammars*). Human languages like English cannot be formally represented using regular or context-free grammars, and more complex representations are required (Chomsky, 1956). To expand on the analogy in Figure 2, human language would be the equivalent of Stravinsky's *Rite of Spring* – or Charlie Parker's *variations* of that piece in *Repetition*.

As one progresses from regular to recursively enumerable grammars, one increases the range of formal languages that can be generated. This is illustrated in Figure 1c. Put otherwise, every regular language is context-free; every context-free language (avoiding issues with empty strings) is context-sensitive; and every context-sensitive language is included in the set of unrestricted (recursively enumerable) languages.

For each type of language there is a corresponding type of computational device capable of recognizing that type of language, as shown in the rightmost column in Table 1. The details of such machines are beyond our scope, but progressively less restricted languages (moving down the rows in Table 1) correspond to machines with progressively more powerful memory mechanisms. For a regular grammar the corresponding finite-state machine has no memory except for the state of the machine, greatly limiting the kinds of computations it can do. For a context-free grammar, the corresponding machine can stack symbols that it encounters "on hold" for later use, giving it more flexibility, which relates to recursion as discussed below. This progression continues until, for unrestricted grammars, the corresponding machine has an infinitely long storage "tape" that supports any computation that can be done by any computer.

We will discuss here whether animal behaviors can be computationally described in terms of regular languages, or if the more complex context-free description is necessary in some instances. In these terms, a behavior or sound sequence may be defined as "syntactic" if its description requires a context-free apparatus. We know of no evidence, or even discussion, that computationally describable behavior in non-human animals may be context-sensitive or more complex.

While admitting that the regular/context-free distinction captures differences between birdsong and human language, Beckers, Bolhuis, Okanoyoa, and Berwick (2012) find the Chomsky Hierarchy misleading: It is too weak in that specific birdsong examples are describable by a narrow class of regular languages; it is too strong in that some aspects of human language are describable in regular terms, while others are more complex. We agree, but our interest is in helping establish what, if any, aspects of human language can be discerned in any other species. The Chomsky Hierarchy is just a useful formal tool in this regard, and asking whether a behavior in a given species is at some level X or a higher level Y that presupposes X seems intrinsically relevant – in particular determining whether a species has the capacity for fully recursive behaviors, as discussed below.

Note, to conclude this section, that application of the Chomsky Hierarchy to formal languages does not require us to be studying language, or for that matter communication. The use of the word "language" in the phrase "formal language" can be misleading. A formal language is simply a tool to describe patterns in sequences in computational terms. Below we discuss what it may mean to use this tool in the study of animal behavior, and if this is testable, what possible implications it may have for comparative psychology.

Ascertaining Context-free Behaviors

When demonstrating that the computational system underlying human language is (at least) context-free, linguists resort to meaning. Consider some example sentences:

(1) a. *Janosz loves Maryzka.*

b. [*The man whose name starts with a "J"*] *loves* [*the woman whose name ends with an "a"*].

c. [*The man whose name* [*that we can't recall*] *starts with a "J"*] *loves* [*the woman whose name* [*that no one can recall*] *ends with an "a"*].

d. [*The man whose name* [*that we can't recall* [*or pronounce*]] *starts with a "J"*] *loves* [*the woman whose name* [*that no one can recall* [*or pronounce*]] *ends with an "a"*].

It is intuitively obvious that one can easily keep complicating the descriptions for the referent of *Janosz* or *Maryzka* in (1), with no principled limit on the complexity of the relevant expressions.

In turn, it is easy to demonstrate through various experiments that the structure in (1) has not changed, at a deep level, from (1a) to (1d): we have made the subject or object of the sentence obviously more complex in each instance, but the parts that we have added have not altered the basic structure, or for that matter the basic thought, that someone named Janosz loves someone named Maryzka (and they have a difficult name, etc.).

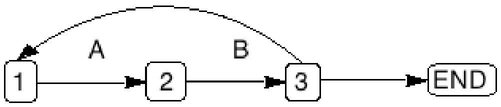
This example illustrates how simple human sentences can present intricate internal structure, such that one can essentially add to their parts (subject, object) without altering their "skeletal" meaning. It is this recursive property that requires a context-free modeling of human sentences. Anything simpler will not allow us to keep adding internal complexity to the constituent parts, *ad infinitum*. Now, it is interesting to observe that the "ad infinitum" assumption is based on the good faith of human testers recognizing one another's intuitions when they say: "I could go on forever making (1) more complex." If we literally could not go on forever, at least in principle, then there would be a trivial way to provide a finite-state representation of the structure in (1) with whatever complexity it has *up to that point*. It should be obvious, for instance, that a dull machine could spell-out each and every letter in this article from beginning to end. It would not be sound to say that such a machine is a useful model of the authors' linguistic behavior, as it would immediately fail if we changed something as minute as *this* word; plainly, a different set of symbols (e.g., "one" instead of "this") would require a totally different machine to recognize the relevant sequence.

There is a more technical way to discuss these issues: in terms of the *generative capacity* of a grammar. Grammatical systems can be studied with respect to their *weak* or *strong* generative capacity. A computational device of the sort just described, which stringed every letter in this article would produce a result that is *weakly equivalent* to how we, the authors, or you, the reader, are generating the present text. However, authors and readers know much more than that. We implicitly know, say, that in this sentence there is a subject (*we*) or a main verb (*know*), and how these relate to each other – well enough, for instance, to also know where adverbials (like *intuitively*) are appropriate or not. Linguists know this as a consequence of having conducted more or less informal experiments to test knowledge of language. A device that just blindly typed each symbol after the other would not possess such knowledge. Chomsky never entertained the use of regular grammars for the characterization of human language precisely because they are too simplistic to achieve adequate structure, i.e., their strong generative capacity is trivial.

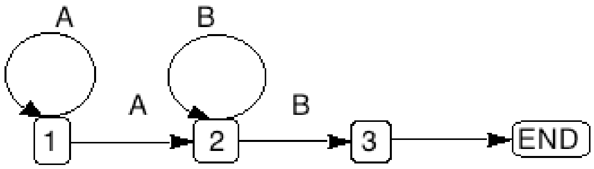
The first problem with examining animal behaviors or signals is that, unlike what happens when examining humans, we don't know what these behaviors mean, if they even mean something, and therefore we cannot agree to just surmise that they can or cannot go on forever in a certain form. More precisely, the problem is with determining the *structure* of an animal's behavior, if it has one. Part of the problem is that the simple experiments we perform with humans to test the strong generative capacity of their grammars ("Do you accept the sentence *John and Mary is friends*?") cannot be performed with animals, since we do not know how to present such ideas to them. More generally, we have no simple way to presume that an animal, after exhibiting a certain complex pattern of the right sort, could go on repeating it or variations of this pattern literally forever – again, in principle, life and memory aside. Interestingly, however, there are indirect ways to evaluate this question given how animals react to alternative sequences of sounds or behaviors.

For example, Gentner et al. (2006) claimed that European starlings (*Sturnus vulgaris*) can learn to classify a sequence of sounds that are produced in two different patterns of the sort $(AB)^n$ vs. A^nB^n. The former means "a number n of pairs AB," while the latter means "a number *n* of A's followed by a number *n* of B's." Importantly, whereas there is a finite-state representation for the first of these patterns (as in (2a)), there is no finite-state representation of the second pattern. The following diagrams illustrate these differences, where directional transitions from states labeled by numbers entail printing symbols *A* or *B* and the procedure ends after hitting the state labeled "END":

(2) a.



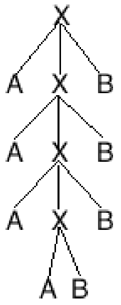
b.



The device in (2a) produces an indefinite number *n* of AB strings, but the device in (2b) does *not* produce an indefinite number *n* of As followed by the same number *n* of Bs. Rather, the device produces *n* as followed by a different indefinite number *m* of Bs (although *n* and *m* could happen to be identical). This is because there is no way to guarantee that the A loop is invoked by the device precisely as many times as the B loop.

Having witnessed the impossibility of representing A^nB^n in finite-state terms, consider, in contrast, how we would achieve the task by using a context-free system (3):

(3) a. X → AXB; **b.** X → AB **c.**



This situation deploys a more complex type of rule, involving an abstract "phrasal" symbol X that has no observable realization. Its sole purpose is to serve as a "derivational support" for a given pattern of As followed by Bs. Now we clearly have the A^nB^n pattern – but rules like (3a) and (3b) are context-free, and the relevant language is no longer regular.

If starlings are able to discriminate $(AB)^n$ patterns as in (2a) from A^nB^n patterns as in (2b), it stands to reason that they are capable of context-free generalizations, or at least as capable as humans in comparable conditions. Here there were no semantic assumptions, just a given structural pattern, which clearly cannot be generated by a finite-state device that the birds allegedly recognize. If indeed the birds are using context-free cognitive operations to achieve their recognition task, they ought to possess the ability to exhibit recursive behaviors – the conclusion is immediate.

But there is a catch: the "and so on" implicit in the rule system in (3) (or the loops in (2)). This distinction is crucial. If our task were to write a finite-state algorithm *for specific values of n* in the A^nB^n pattern, this would be trivial. One could simply assume a finite-state algorithm that writes *ab*, another one that writes *aabb*, a different one that writes *aaabbb*, up to as many of these combinations as we happen to have observed. In the case of the starlings, three such iterations were observed, but even if it were four, five or one thousand, this would make no difference. What would make a difference is having in principle *any* relevant combination of the right sort.

This is at the crux of much recent discussion to figure out whether the starlings may have identified the relevant structures as in (4a) with the reasoning in (4b):

(4) a. ab, aabb, aaabbb

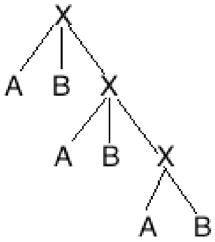
b. 1 "a" followed by 1 "b"; 2 "a's" followed by 2 "b's"; 3 "a's" followed by 3 "b's"

Evidently this reasoning could not go on unless one knows how to count (a recursive procedure). Studies summarized in Hauser (2000) suggest that many animals, like human infants (Gelman, Meck, and Merkin, 1986), understand the concept "small number" as a pattern of some sort – although apparently not the concept of "exact numerosity" as numbers grow minimally large. In this view of things the starlings would start failing the recognition task as the number *n* of symbols grows to four, five, and so on (This skeptical analysis is challenged in a new experiment by Abe and Watanabe (2011), although Beckers et al. (2012) argue that the design of that study is inadequate). Here, we are not attempting to take a position on whether the starlings succeed or fail in recognizing recursive patterns. Our goal is purely methodological, attempting to provide ways of ascertaining these claims.

It is also important to recognize that establishing the presence of recursion in a system by merely examining the system's outputs is difficult in one other regard. Imagine that, by some reasonable method, we had convinced ourselves that a given behavior does go on forever, at least in principle. Even then we may not yet have demonstrated true recursion as the hallmark of a context-free system, and we may instead be witnessing a lesser form of "iteration" that can be modeled in finite-state terms. While both iteration and recursion handle repetitions, they do so very differently. Iteration involves a loop as in (2b), which causes the relevant repetition by literally going back to a point in the derivational flow. By inserting one such loop in the preceding sentence we could generate "going back to a point in the derivational flow derivational flow derivational flow…" While repetitive, the resulting language is not recursive in the sense that interests us here. Essential to true recursion is that a given pattern X is identified and another instance X' of the very same pattern is produced within X, for example as in rule (3a). In that rule "self-referentiality" is direct (the rule contains in its right portion the very symbol in the left); the process can be indirect also, so long as somewhere in the derivational flow another instance of X is necessarily invoked by a rule or set of rules that the initial X triggers. Now as it turns out, if we simply examine a string with a given characteristic repetition it is hard to tell whether the repetition is genuinely recursive or merely iterative.

The difficulty is easy to illustrate with the following rule system, which generates the formal language $(AB)^n$ in a structurally more powerful way than the mechanism in (2a):

(5) a. X → A B X; **b.** X → A B **c.**



Since every regular language is a context-free language, just as we can provide a finite-state representation of $(AB)^n$ as in (2a), we can also provide the more complex representation in (5c). This object presents what is commonly called "tail recursion," a form of recursion that is weakly equivalent to "iteration" by way of a finite-state loop. The observation of a behavior (in this instance the pattern $(AB)^n$) that could be described in terms of "tail recursion," unbounded in length as it may be, is clearly not enough to surmise an underlying context-free generative device.

Only in the *simultaneous and parallel* presence of "tail recursion" and "head recursion" – as implicit in the set of sentences in (1) or objects of the form A^nB^n – is full recursion present. This is important because witnessing mere repetitions, even if they could (in principle) go on forever, will simply not suffice to establish that a given behavior is plausibly recursive and that, therefore, it presupposes a context-free device in some form. Experimenters will need, instead, to observe or induce behaviors that cannot be explained away as iterative. We return to that complex issue below, and offer some suggestions.

Note, also, that the two "methods" of demonstrating full recursion presented in this section are very different. The fully recursive nature of a language A^nB^n stems from the structural fact that this particular pattern (if *n* is not bounded) cannot be generated in finite-state terms, given its internal symmetry. In contrast, the method to ascertain the fully recursive character of the structures in (1) is essentially semantic, again assuming one can go on forever adding complexity to subject and predicate. In what follows we explore other putative ways to ascertain full recursion that may be useful for comparative studies.

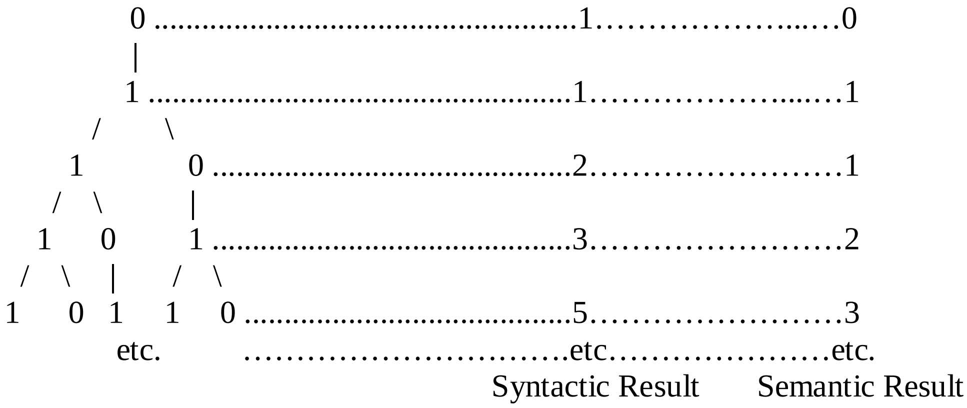
Fibonacci Grammars

We next consider a sequence pattern that, while relevant to the computational complexity that separates regular from context-free grammars, addresses some of the methodological concerns raised in the previous section. The only way one can test whether a given animal recognizes a language of the sort A^nB^n is by giving the animal progressively more complex instances of a mirror symmetry – where inevitably an upper limit is reached. In the pattern we present next, the symmetry is more intricate, and it does not depend on string length. In fact, relevant sequences can be indefinitely long, and what emerges in them is a “rhythm” – if they are recognized. We propose that this approach overcomes some of the computational issues raised above, while at the same time provides a method for testing any species that can be trained to discriminate sound sequences, e.g., songbirds, perhaps some bats (e.g., *Tadarida brasiliensis*; Bohn, Schmidt-French, Schwartz, Smotherman, and Pollak, 2009), or even hyrax (Kershenbaum, Ilany, Blaustein, and Geffen, 2012).

An interesting extension of Chomsky-style rewrite rules proposed by Lindenmayer (Prusinkiewicz and Lindenmayer, 1990) allows for the generation of famous mathematical sequences of the sort widely observed in nature. In Lindenmayer-systems all applicable rewrite rules of the sort discussed above apply simultaneously to a given derivational line, and there is no distinction between terminal and non-terminal symbols. While Chomsky’s grammars advance symbol by symbol, Lindenmayer’s grammars advance derivational line by derivational line. This has interesting consequences when the grammar is not given any specific termination limit: It can stop at any given generation (at any given line). At that point the device can be used to model plant branching or shell formation. For instance, the context-free rule system in (6) generates (7):

(6) 0 → 1 , 1 → 1 0

(7)



The number of symbols in each derivational line yields the Fibonacci sequence (1, 1, 2, 3, 5 ...), both in syntactic terms (counting each symbol in every generation as a unit) and in semantic terms (adding up the arithmetic value of each symbol as either “one” or “zero”).

A “phrasal” object of the sort in (7) can be synthesized, at any given derivational line, in terms of recognizable sounds (by making the *0* stand for some particular sound, for instance *bi*, and *1* stand for a different sound, say *ba*). This is what Saddy (2009) did in order to investigate the reaction of humans to pseudo-random strings of syllables for which there are no simple statistical regularities at the level of their neighboring occurrences, one after the other. Subjects of the experiment hear synthetically generated sequences like *ba-bi-ba-ba-bi* (corresponding to the sequence of *1*'s and *0*'s in the bottom line of the tree in (7)). The local transition probabilities between symbols in strings generated by these grammars are close to random: As the graph grows one cannot tell, when processing a *ba*, whether the next element is going to be a *bi* or another *ba*.

In the experiment, the pseudo-random strings are compared to others that lack the ordering implicit in the context-free regularities in (6). To construct test cases, a random string of ones and zeroes is created; then different substitutions of these symbols for string bits in the *ba/bi* output of the L-grammar are inserted into the random string – to make the strings comparable – yielding versions of the experiment. Subjects are asked to listen to 3-minute long strings of the Fibonacci strings. After the training phase, they are presented with pairs of candidate strings lasting 10 seconds each, and they are asked to indicate which of the pair is most similar to the training set. Although subjects are not perfect in their recognition, the percentage of times they discriminate the Fibonacci option from the quasi-random option is significantly above chance. In all probability, humans are identifying *constituents* in the string of syllables, at a higher level of abstraction than the mere list of signals heard. In other words, humans may be using their context-free linguistic abilities to appropriately parse these context-free, non-linguistic, objects.

Can a version of Saddy's experiment be performed with birds, bats, or other vocal learners? As Suge and Okanoya (2010) observe, Bengalese finches seem to have an ability to construct and perceive "acoustic phrases." In a famous experiment performed by Fodor, Bever, and Garrett (1974), human subjects given a "click" sound stimulus in the middle of a phrase displace the perception of the sound to the phrasal edge. In other words, when actually hearing "the phrasal CLICK edge," humans perceive the event as "the phrasal edge CLICK." Suge and Okanoya (2010) found that Bengalese finches react the same way when presented with natural sequential chunks of birdsong. This result raises the issue of whether the birds are literally taking the units in point as phrases (generated by a context-free grammar) or, rather, as Berwick, Okanoya, Beckers, and Bolhuis (2011) carefully study, they may be using a less powerful regular grammar.

Berwick et al. (2011) informally speak in terms of "birdsong syntax," although they also come shy of arguing that this syntax is context-free in any of the bird species they have studied. Tu and Dooling (2012) studied the sensitivity of budgerigars to the order in which naturally occurring elements within the warble are presented, as compared to canaries and zebra finches. While the latter species performed at chance in identifying partially scrambled sequences, the budgies were not "fooled" – except if *all* the relevant ongoing warble stream elements were randomly presented. Tu and Dooling (2012) see in this "sensitivity to sequential rules governing the structure of their species-specific warble songs" (p. 1151). They suggest that the observed behavior points to a "rule that governs the sequential organization of warble elements in natural warble song and is perceptually salient to budgerigars but unavailable to the other two species" (p. 1158).

In our opinion, an important issue is not just whether the song is rule-governed, but rather whether the rule is context-free – what we are calling "syntactic." We suspect that budgerigars, with their sensitivity to rather elaborate sequential behaviors, would be good subjects for a version of Saddy's experiment, perhaps after being trained in the recognizing techniques discussed by Tu and Dooling (2012).

To conduct this experiment, where the synthesized grammar sketched above uses *ba* and *bi*, two different forms of the budgerigar warble would need to be used. The experimental question would then be to determine whether trained budgerigars could discriminate random from Fibonacci patterns, and if so with what accuracy. As in the Gentner et al. (2006) or Abe and Watanabe (2011) experiments, this example uses artificial grammars without semantics, thus it gives the animal a fair chance at succeeding in the recognition task. Moreover, this particular pattern is not limited to expressions of the A^nB^n sort, whose phrasal representation depends on establishing that *n* has no upper limit. Instead, any isolated portions of the Fibonacci grammar could be presented to the bird, for as long as necessary. Humans apparently recognize the relevant "rhythm" in a matter of seconds after very little training and, as noted, the length of the training or exposure phases has no correlation with the identification of the pattern. We suggest that this approach has great potential to ascertain if an animal can recognize strings that only context-free systems generate.

Grammar in Thought

There is no principled reason why complex cognition in animals must be limited to communication or, for that matter, reactions to human-made sound sequences. At some level, animals must think (Griffin, 1984; Hauser, 2000), and their ability to solve problems may well require computational capacities (Gallistel and King, 2009). Whether such thoughts, however, are sufficiently complex to merit the qualification of "syntactic" within the Chomsky Hierarchy is an open question that deserves careful consideration.

Examples of remarkable animal behaviors that could require complex computational abilities occur in a variety of species. For example, New Caledonian crows are well known for manufacturing several different types of tools out of twigs and leaves, each with a separate function (Hunt, 1996; Hunt, Corballis, and Gray, 2006) and have been observed using one tool to retrieve another (Taylor, Hunt, Holzhaider, and Gray, 2007). Even more diverse and complex tool use has been described for bearded capuchin monkeys (Mannu and Ottoni, 2009) and common chimpanzees (Limongelli, Boysen, and Visalberghi, 1995; Sanz and Morgan, 2010). Baboons apparently know the rank and kinship of every individual in their group and behave selectively as a consequence (Cheney and Seyfarth, 2007). Lions (Heinsohn and Packer, 1995; Stander, 1992), wild dogs (Creel and Creel, 2002), and Harris' hawks (Bednarz, 1988) all capture prey by hunting in groups with individuals frequently adopting specific roles. These are the sorts of examples that have led Gallistel, most notably, to openly argue for a computational mind in non-human animals.

However, the non-trivial issue in all these instances is whether we are witnessing (a) a cognitive analysis of a situation, of the complex computational sort or, rather, (b) a predictable sequence of behaviors that through trial-and-error happen to have been combined by chance and are now repeated in response to the appropriate stimulus. Even humans surely present complex behaviors whose bases are arguably not cognitive in any interesting sense. Mechanically brushing one's teeth or patting one's head (among other instances that easily come to mind) are not obviously complex computational actions.

Intuitively, one way to determine if complex cognitive ability is involved is to see whether the behavior in question is *creative*, a term we are using in a very specific sense here. The typical, and in fact inversely related, dimensions along which "creativity" varies are plasticity and boundedness. The more genuinely plastic a behavior is, the less of a predictable end it has: Ideally it should be able to go on changing forever, at least in principle. Conversely, if a behavior has a predictable end, there is a sense in which those are the boundaries within which it repeats itself. Of course, endless plasticity would also occur if a given organism's mind were unbounded, for instance presenting indefinitely many ideas. The computational theory of mind, however, presupposes that animal or, for that matter, human minds are bounded. This is precisely where it matters whether a presupposed computational procedure should be context-free, as only this sort of computation (or more complex ones) achieves full recursion – and with it, general plasticity as an observable side effect – in spite of its intrinsic boundaries. In effect this illustrates the Cartesian ideal of creativity: unbounded behavior given bounded resources.

Whereas an underlying recursion can provide behavioral plasticity, the presence of plasticity does not prove full recursion. Only in formal or quasi-formal systems (e.g., math, language, music) can one determine, beyond reasonable doubt, that the system observed is fully recursive, and thereby creative in the sense in which we are using the term. The very intricate formal structure of the (right sort of) system allows us to conclude by induction that if a pattern holds at some point, then it will hold for the entire system. Alas, life (nature, the observable universe, animal behaviors…) is not that well behaved. Neither are humans, but here, semantics allows us to conclude "and so on." How to allow ourselves to conclude "and so on" in the case of animal behaviors is rather more nuanced. In the case of the Fibonacci grammars, their very structure allows us to conclude their recursion. But what about observed behaviors?

That complex matter is all the more challenging because, as noted at the end of section 3, some patterns of simple repetition are actually irrelevant ("tail/head" recursion, weakly equivalent to iteration, vs. "full" recursion), and thus we need to distinguish between them. In the following, we illustrate how context-free behavior might hypothetically be recognized by contemplating a purely fictitious example that was inspired by observations of burrowing owls (*Athene cunicularis*) using manure to line their burrows (Smith and Conway, 2007). Please note that our intent is explicitly not to claim that owls use complex cognition. Rather, we would like to show the kind of processes that must be present to plausibly reach such a conclusion for any behavior, whether it is tool use or cooperative hunting, among other possible complex behaviors that animals exhibit.

Assume that burrowing owls systematically find insects under dung. This sort of behavior could be systematized in Aristotelian propositional terms as follows (simplifying the formalism, explicitly disregarding nuances about pronouns and the like):

(8) i. Premise *p*: You lift dung.
ii. Premise *q*: Insects thrive there.

It should be easy to see that an inference *p* → *q* can be implemented in finite-state terms of the sort in (1), by having a system with two states, each looped, the first loop printing "*p*" and the second loop printing "*q*." (Associating the loop to a printing of the representation for a premise entails that the iterated premise could be printed many times – but we take that to have no logical consequence, since the premise would simply reassert itself.) Then the system halts at the END state. In turn, we could interpret the transitions as inferential. Interestingly, burrowing owls often find dung and bring it close to their burrow. Eventually the dung fills up with insects, and the rest of the reasoning follows. Codifying that behavior requires new Aristotelian premises, as follows:

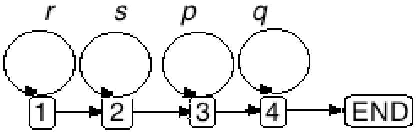
(9) i. Premise *r*: You find wet dung.
ii. Premise *s*: You bring wet dung to burrow.

Intuitively we want *r* → *s* to feed the logical transition *p* → *q*:

(10) a. (*r* → *s*) → (*p* → *q*)
b. If (if you find wet dung you bring it to the burrow) then (if you lift it insects thrive there)

This inference could be expressed in finite-state terms as in (9), where again the premises are conventionally printed in iteration loops at given states and inferential rules correspond to state transitions (logically the parentheses in (8) play no role in the reasoning):

(11)



And so on: We could complicate these causal inferences as much as we want. Surely this situation of iteration/"tail recursion" could also be expressed in context-free terms: A system including the following rewrite rules yields the same strings as (11):

(12) S → A B C D; A →r A; B → s B; C → C p; D → D q; A → r; B → s; C → p; D → q

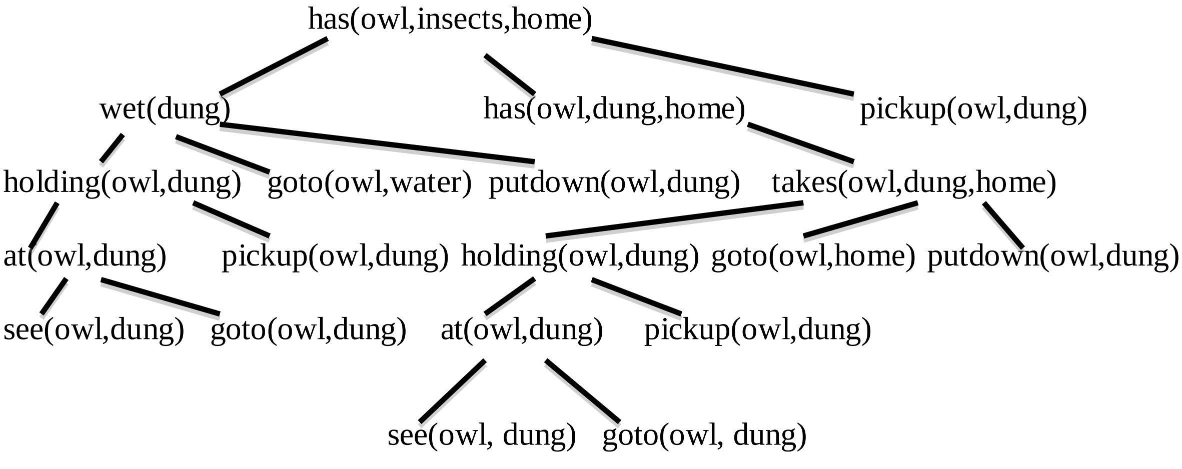
However, inasmuch as the simpler algorithm in (11) also works for the output string we are interested in, we cannot conclude that the more complex structure is presupposed. In other words, even if we were to observe this behavior, and we were to analyze it logically as in (10), after convincing ourselves that these particular decisions are appropriate, we would still not have demonstrated the presence of a recursive behavior.

This in turn means that, in order to *ascertain* that the observed behavior requires a more complex sort of computational machinery – one that could induce full recursion – we have to observe or induce a more complex related behavior. To illustrate this methodological point we present the following hypothetical scenario in terms of a set of (simplified) first-order predicate logic assertions. Variable terms, such as the one implicit for “who,” are capitalized and universally quantified; constants are lower case. Each line is effectively an if-then rule, where implication is indicated by “→” and conjunction by “∧.” For expository reasons, we are simplifying, making the extra-logical assumption that rule antecedents are ordered left to right, temporally. (To make the inference valid we should add quantification over events, and have each predicate below include one more argument: an event variable; then the reasoning could be temporally articulated without ad hoc assumptions. We avoid this complication so as not to cloud the picture in this context.)

(13) wet(dung) ∧ has(Who,dung,home) ∧ pickup(Who,dung) → has(Who,insects,home)
holding(Who,What) ∧ goto(Who,water) ∧ putdown(Who,What) → wet(What)
takes(Who,What,home) → has(Who,What,home)
see(Who,What) ∧ goto(Who,What) → at(Who,What)
holding(Who,What) ∧ goto(Who,Where) ∧ putdown(Who,What) →
takes(Who,What,Where)
at(Who,What) ∧ pickup(Who,What) → holding(Who,What)

The second rule above might be expressed in English as: “If an animal (Who) is holding an object (What) and the animal goes to water and puts the object down, then the object will become wet.” The first goal of our exercise is to prove the assertion: has (owl, insects, home). Predicates *pickup*, *putdown*, *see*, and *go to* are assumed to be primitive actions an animal can perform (therefore not defined by the rules in the reasoning). (14) shows what “backwards chaining” of the rules gives us, starting with the goal:

(14)



The second goal of the exercise stems from observing the “tree” in (14). Clearly it branches in more than one side. Now *if this branching could go on*, then the relevant underlying structure, if modeled computationally, could not be a regular language.

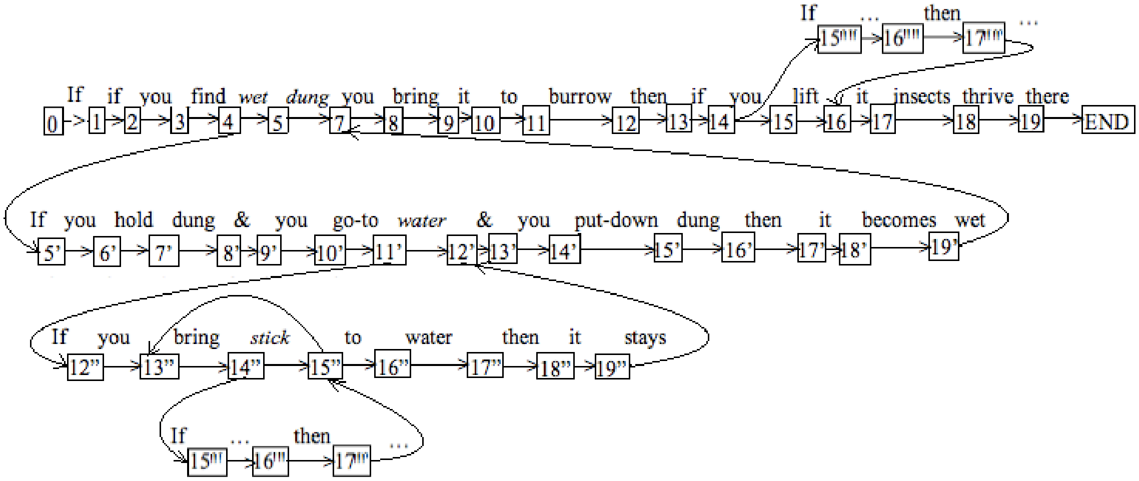
An immediate difficulty in constructing relevant examples or observing relevant behaviors arises in terms of determining that the reasoning on either complex branch could, indeed, go on forever. In the thought experiment we complicated the task of bringing dung to the burrow by introducing a subroutine of requiring it to be wet. Experimenters could build conditions, for this scenario, of how the animal would react to difficulties in getting the dung wet. And if this requires, say, building a dam in a water stream, what happens if the materials for the dam are not readily available – and so on. So the logical structure in (10) would effectively be complicated as in (15), as needed (where each added line represents a logical response to a difficulty added to the task, italicized in each instance):

(15)
If (if you find *wet dung* you bring it to the burrow) then (if you lift it insects thrive there)
↑
If you hold dung & you go-to *water* & you put-down dung, then it becomes wet.
↑
If you bring *stick* (bring stick, bring stick…) to water, then it stays.
↑
If … then …

The experimenter needs to at some point decide "when enough is enough" to confidently say that relevant inferences (to resolve particular difficulties as they emerge) can be added upon indefinitely. We do not have anything meaningful to say here about this problem, other than it also arises for humans and, as noted, we decide on it by common agreement. We submit that if an animal keeps solving problems in a logical chain as in (15), with creativity appropriate to the specific task as opposed to some fixed set of responses - that constitutes prima-facie evidence of the logical chain that would be interesting to detect.

The more modest point we are attempting to make, however, is that – in order for it not to be dismissed as possibly being of the iterative sort – the "and so on" (however it is established) would have to be present on at least two sides of a "tree" of the sort in (14), which means at least two separate inferential chains need to be invoked as the behavior is observed. Furthermore, even if two relevant, open-ended, inferential chains are observed, it would be important to ascertain that neither of them reduces to a finite-state subroutine with trivial looping in the derivational flow. In other words, it must be kept in mind that even a behavior as complex as the one we are attempting to characterize may well have an (irrelevant) characterization as in (16) (where this time we have chosen to represent each putative logical element as a transitional state):

(16)


If … then …
15'''' 16'''' 17''''
If if you find *wet dung* you bring it to burrow then if you lift it insects thrive there
0 1 2 3 4 5 7 8 9 10 11 12 13 14 15 16 17 18 19 END
If you hold dung & you go-to *water* & you put-down dung then it becomes wet
5' 6' 7' 8' 9' 10' 11' 12' 13' 14' 15' 16' 17' 18' 19'
If you bring *stick* to water then it stays
12" 13" 14" 15" 16" 17" 18" 19"
If … then …
15''' 16''' 17'''

The key in designing appropriate experiments would be to determine if such a finite-state analysis is sufficient or insufficient. The structural approach presented in the foregoing sections capitalized on paired occurrences of two different loops in various states in the finite-state chain, which is impossible to obtain in an object with the simplicity of (16). The semantic approach, in contrast, is less definitive: It builds on the intuition that, when facing new problems (dung that happens not to be wet, water that happens not to be contained) the animal does not just go into a subroutine that blindly loops miraculously appropriate behaviors to address the particular problem in point. Rather, the animal in those circumstances analyzes the situation and organizes its thought process in ways consistent with a logic structure as in (14).

Ascertaining whether the latter is the case will not be easy. One way may be to observe whether the new behavior in point (presumably induced by the experimenter) is something that the animal repeats without apparent reason or instead is a clear response to the situation. In the example discussed, in order to get an object wet all that is needed is to drop it in water once. If the object is dropped and picked up and dropped again and so on, this may be an indication that the behavior does not constitute an analysis of the situation. Alternatively, the behavior in point may well be *appropriately* not context-sensitive. In the thought experiment above, for example, building a dam to aid in the collection of water clearly involves repetition of a task (bringing sticks and placing them). Repetitiveness in that task would be appropriate, while repetitiveness, say, in building the entire dam would not – at least not from the semantic perspective of the human experimenters.

In our view, "creative behavior" in the sense we are interested in here is arguably a combination of plasticity and recursion, potentially yielding the flexibility characteristic of animals noted for their problem-solving abilities. Full recursion as such proves creativity in this sense, but it is hard to find other than in formal systems. The characteristic novelty of plasticity, if couched within a carefully constructed experiment that builds it on two (or more) separate sides of a logically connected inference chain of the sort above, would provide insight into the cognitive requirements of the behavior. Many interesting experiments with animals do appear to show novelty in behavior, but we have not seen any case that controls for the formal nuances we have presented here, which we believe would make a much stronger case for full recursion, and with it the possibility that an animal mind does involve context-free rule-governed behavior, or in our terms is "syntactic." Showing this, in turn, would provide evidence that a key step toward the evolution of language exists in a nonhuman animal, something that has yet to be accomplished.

A Role for Computational Modeling

Computer simulations based on multi-agent systems in which individuals are simulated as they learn/evolve communication systems (Cangelosi and Parisi, 2002; Lyon, Nehaniv, and Cangelosi, 2007) can help study the origins of language. Results, however, have been limited when it comes to providing a compelling explanation of emergent language – or even the simpler question of the relationship between the Chomsky hierarchy and animal signaling systems.

Consistent with the concept of a hierarchy of levels as above, we have previously viewed computational models of emergent communication in terms of the levels of complexity that they involve (Wagner, Reggia, Uriagereka, and Wilkinson, 2003). The simplest, lowest level models involve evolving communication using single unstructured tokens – something "below" the Chomsky hierarchy. Models of this sort have used agents (simulated animals) situated in an external environment, typically an artificial world in which the agents move throughout the landscape (e.g., Marocco and Nolfi, 2007). Motivated by observations of animal communication (e.g., alarm calls) such studies address the issue of how grounded signals (linked to external entities) could arise via evolution. Agents in these simulations have tasks to perform, such as finding food/mates, avoiding predators, or trading resources (Wagner and Reggia, 2006). It is agent performance on these tasks (not their communication abilities directly) that determines their reproductive fitness.

In one typical study (Reggia, Schulz, Wilkinson, and Uriagereka, 2001), evolutionary and ecological factors were systematically manipulated to determine when food or alarm calls would evolve. Findings included that: (i) a sufficient density of agents is necessary for calls of any type to evolve, (ii) food calls evolve most frequently when food sites were large but few in number, and (iii) alarm calls evolve very easily even when only a few predators are present in the simulated world.

A higher level of complexity has been examined in simulations examining the emergence of structured communication acts. Most such studies have been with non-situated agents where adaptation involved only learning, not evolution (reviewed in Wagner et al, 2003). Some simulations have used evolution with/without learning, demonstrating, aside from mechanisms by which spatial and temporal variation in communication can arise, that (i) evolution-plus-learning can be more effective than either one alone, and (ii) structured signals used for cooperation can evolve when the number of situations to communicate is larger than the repertoire of signal components (Kirby and Hurford, 1997; MacLennan and Burghart, 1993; Werner and Todd, 1997).

Few simulations have examined situations most relevant to language origins: the evolution of structured communication among situated agents. In one study, agents controlled by neural networks evolved to perform appropriate actions in a world of both edible and poisonous food sources, and then subsequently learned to communicate via appropriate utterances (Cangelosi, 2001). The communication system that emerged was in many cases structured, but very limited. Based on examining the evolved agent population, the investigators characterized the agent "language" that emerged as consisting of a set of fewer than 10 "noun-verb pairs," such as "avoid plant A" or "approach plant B," and concluded that the Baldwin effect was in play (Munroe and Cangelosi, 2002). This (regular) language would of course be trivial to describe with a finite-state grammar.

In sum, it has proven difficult so far to model the emergence of even the simplest languages in simulations of agent populations. Given our analysis in the preceding sections, a logical approach would be to expand future computational work to characterize *the evolution of non-linguistic sequential behaviors* in terms of the Chomsky hierarchy, something that will probably be easier to model than anything communicative. Such modeling would be most effective if it can be tied to hard data from experimental studies, perhaps along the lines suggested in the two preceding sections. For instance, if carefully constructed experiments with primates argue for behaviors that would be reasonably described as context-free, a simulation with artificial agents in the observed conditions may perhaps be shown to lead to the emergence of the relevantly structured computation. Simulations may shed light on any processes or behaviors that facilitate the emergence of sequential behaviors, plausible intermediate stages on a path from such mechanisms to language, social factors involved in the acquisition of language, and more.

Conclusions and Future Research

The theoretical approach we have outlined here provides a method for designing experiments that have the potential to reveal the computational requirements needed for different forms of behavior. Intuitively, a genuinely creative behavior is different from any other, enough for it not to have resulted from instinct or trial-and-error. The hallmark of individual cognition, creativity separates what one does when, say, brushing one's teeth from what one attempts when the toothpaste is finished or the brush breaks. The computational modeling of such behaviors allows us to classify them rationally, according to the sorts of rules that would allow a logical machine to perform them, or how much memory relevant algorithms would require to carry out such rules. Even at the lower levels of the ensuing hierarchy interesting questions arise about just how plastic relevant behaviors are and when they start being genuinely creative.

We have concentrated on behaviors that we have called "syntactic" in that they are, effectively, plastic and open-ended. Rules that lead to such behaviors are called context-free within the Chomsky Hierarchy, and they exhibit full recursion. An interesting, and difficult, question is whether full recursion is uniquely human or is present in other animals. We have suggested two different ways of going about answering this question: one structural and one semantic.

The structural approach builds on the symmetry that Fibonacci patterns present, which can be constructed by way of a simple rewrite system. Testing whether a subject can parse the output of these grammars, appropriately encoded in terms of relevant signals that an animal may recognize, does not require training with odd patterns of unreasonable size. Thus, such experiments should be possible to conduct with any outcome providing valuable information.

The semantic approach was proposed as an observational exercise involving complex logical chains, crucially on at least two separate "tree branches," thereby not being reducible to finite-state analysis. Relevant experiments would have to wrestle with knowing when to declare a relevant logical chain successful, regardless of how many intermediate distractions or additional subroutines need to be added to keep the subject on task. Nevertheless, again putative results from such experiments would be extremely informative, and they have the added virtue that they need not involve signals or discrimination among artificial grammars.

The experimental program we have outlined has relevance to other cognitive tasks. For example, we noted in passing in section 2 that one way to characterize differences between the various levels in the Chomsky Hierarchy is in terms of the corresponding algorithms that would recognize formal languages at each level, and the amount and sort of memory that each requires. This invites questions about whether that abstract memory could have anything to do with memory in a psychological sense. The hypothesis is certainly not logically required, but it is intriguing, and could suggest the possibility that forms of memory play a role in allowing animals' access to progressively more complex structures within the Chomsky Hierarchy.

This last point could also be approached from a neurobiological perspective. At present, it is not well understood how memory is encoded or how gene expression influences that process. Nonetheless, recent gene expression profiling, as well as manipulation of single genes, in birds and mammals has provided insight into the genetic basis of learning and memory. Such results raise the question of whether manipulation of specific genes could alter the ability to perform complex computational tasks. We submit that the program we have sketched here for the study of language and broadly related behaviors provides a feasible way to test hypotheses and potentially gain insight into the origins of creative thought.

Acknowledgements: We thank J. Vonk and T. Shackelford for the invitation to contribute to this special section and to two anonymous reviewers for their comments. GSW's research is supported by NSF DEB-0952260.

Received 26 September 2012; Revision submitted 12 December 2012; Accepted 15 December 2012

References

Abe, K., and Watanabe, D. (2011). Songbirds possess the spontaneous ability to discriminate syntactic rules. *Nature Neuroscience, 14*, 1067-1074.

Beckers, G. J. L., Bolhuis, J. J., Okanoyoa, K., and Berwick, R. (2012). Birdsong neurolinguistics: Songbird context-free grammar claim is premature. *Neuroreport,* *23*, 139-145.

Bednarz, J. C. (1988). Cooperative hunting Harris' hawks (*Parabuteo unicinctus*). *Science, 239*, 1525-1527.

Berwick, R. C., Okanoya, K., Beckers, G. J. L., and Bolhuis, J. J. (2011). Songs to syntax: The linguistics of birdsong. *Trends in Cognitive Science, 15*, 113-121.

Bohn, K. M., Schmidt-French, B., Schwartz, C., Smotherman, M., and Pollak, G. D. (2009). Versatility and stereotypy of free-tailed bat songs. *PLoS One, 4,* e6746.

Cangelosi, A. (2001). Evolution of communication and language using signals, symbols and words. *IEEE Transactions on Evolutionary Computation, 5,* 93-101.

Cangelosi, A., and Parisi, D. (Eds.) (2002). *Simulating the evolution of language*. New York: Springer.

Cheney, D. L., and Seyfarth, R. M. (2007). *Baboon metaphysics: The evolution of a social mind.* Chicago: University of Chicago Press.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory, 2*, 113-124.

Creel, S., and Creel, N. M. (2002). *The African wild dog: behavior, ecology and conservation.* Princeton: Princeton University Press.

Fodor, J. A., Bever, T. G., and Garrett, M. F. (1974). *The psychology of language.* New York: McGraw-Hill.

Gallistel, C. R., and King, A. P. (2009). *Memory and the computational brain: Why cognitive science will transform neuroscience*. New York: Wiley-Blackwell.

Gelman, R., Meck, E., and Merkin, S. (1986). Young children's numerical competence. *Cognitive Development, 1*, 1-29.

Gentner, T. Q., Fenn, K. M., Margoliash, D., and Nusbaum, H. C. (2006). Recursive syntactic pattern learning by songbirds. *Nature*, *440*, 1204-1207.

Griffin, D. R. (1984). *Animal thinking*. Cambridge: Harvard University Press.

Hauser, M. D. (2000). *Wild minds: What animals really think.* New York: Henry Holt and Company.

Hauser, M. D., Chomsky, N., and Fitch, W. T. (2002). The faculty of language: What is it, who has it, and how did it evolve? *Science*, *298*, 1569-1579.

Heinsohn, R., and Packer, C. (1995). Complex cooperative strategies in group-territorial African lions. *Science 169*, 1260-1262.

Hunt, G. R. (1996) Manufacture and use of hook-tools by New Caledonian crows. *Nature, 379*, 249-251.

Hunt, G. R., Corballis, M. C., and Gray, R. D. (2006). Design complexity and strength of laterality are correlated in New Caledonian crows' pandanus tool manufacture. *Proceedings of the Royal Society of London B, 273,* 1127-1133.

Jurafsky, D., and Martin J. (2000). *Speech and language processing*. New York: Prentice Hall.

Kershenbaum, A., Ilany, A., Blaustein, L., and Geffen, E. (2012). Syntactic structure and geographical dialects in male rock hyraxes. *Proceedings of the Royal Society of London B, 279*, 2974-2981.

Kirby, S., and Hurford, J. (1997). Learning, culture and evolution in the origin of linguistic constraints. In P. Husbands and I. Harvey (Eds.), *Proceedings of the Fourth*

European Conference on Artificial Life (pp. 493-502). Cambridge: MIT Press.

Limongelli, L., Boysen, S. T., and Visalberghi, E. (1995) Comprehension of cause-effect relations in a tool-using task by chimpanzees (*Pan troglodytes*). *Journal of Comparative Psychology, 109,* 18-26.

Lyon, C., Nehaniv, C., and Cangelosi, A. (Eds.) (2007). *Emergence of communication and language*. New York: Springer.

MacLennan, B., and Burghardt, G. (1993). Synthetic ethology and the evolution of cooperative communication, *Adaptive Behavior, 2*, 161-188.

Mannu, M., and Ottoni, E. B. (2009). The enhanced tool-kit of two groups of wild bearded capuchin monkeys in the Caatinga: Tool making, associative use, and secondary tools. *American Journal of Primatology, 71*, 242-251.

Marocco, D., and Nolfi, S. (2007). Communication in natural and artificial organisms: Experiments in evolutionary robotics. In C. Lyon, C. Nehaniv, and A. Cangelosi (Eds.), *Emergence of communication and language*. New York: Springer (pp. 189-205).

Munroe, S., and Cangelosi, A. (2002). Learning and the evolution of language. *Artificial Life, 8*, 311-339.

Prusinkiewicz, P., and Lindenmayer, A. (1990). *The algorithmic beauty of plants.* New York: Springer-Verlag.

Reggia, J., Schulz, R., Wilkinson, G., and Uriagereka, J. (2001) Conditions enabling the emergence of inter-agent signalling in an artificial world. *Artificial Life, 7*, 3-32.

Saddy, D. (May, 2009). *Perceiving and processing recursion in formal grammars*. Paper presented at the Recursion: Structural Complexity in Language and Cognition Conference at the University of Massachusetts, Amherst.

Sanz, C. M., and Morgan, D. B. (2010) The complexity of chimpanzee tool-use behaviors. In E. V. Lonsdorff, S. R. Ross, and T. Matsuzawa (Eds.), *The mind of the chimpanzee: Ecological and experimental perspectives* (pp. 127-140). Chicago: University of Chicago Press.

Smith, M. D., and Conway, C. J. (2007). Use of mammal manure by nesting burrowing owls: A test of four hypotheses. *Animal Behaviour, 73,* 65-73.

Stander, P. E. (1992). Cooperative hunting in lions: The role of the individual. *Behavioral Ecology and Sociobiology, 29,* 445-454.

Suge, R., and Okanoya, K. (2010). Perceptual chunking in the self-produced songs of Bengalese finches (*Lonchura striata var. domestica*). *Animal Cognition*, *13,* 515-523.

Taylor, A. H., Hunt, G. R., Holzhaider, J. C., and Gray, R. D. (2007). Spontaneous metatool use by New Caledonian crows. *Current Biology, 17,* 1504-1507.

Tu, H. W., and Dooling, R. J. (2012). Perception of warble song in budgerigars (*Melopsittacus undulates*): Evidence for special processing. *Animal Cognition, 15,* 1151-1159.

Wagner, K., and Reggia, J. (2006). The emergence of an internally grounded multireferent communication system. *Interaction Studies, 7*, 105-129.

Wagner, K., Reggia, J., Uriagereka, J., and Wilkinson, G. (2003). Progress in the simulation of emergent communication and language. *Adaptive Behavior, 11,* 37-69.

Werner, G., and Todd, P. (1997). Too many love songs: Sexual selection and the evolution of communication. In P. Husbands and I. Harvey (Eds.), *Proceedings of the Fourth European Conference on Artificial Life* (pp. 434-443). Cambridge: MIT Press.