Comparative evolutionary approaches to language: on theory and methods

Jeffrey Watumull, Marc Hauser & Robert C. Berwick

Abstract.

Several theoretical proposals for the evolution of language have sparked a renewed search for comparative data on human and nonhuman animal computational capacities. However, conceptual confusions apparently still plague the field, leading to experimental evidence that fails to test for comparable human competencies. We focus on Rey and colleagues' (2012, *Cognition*) artificial language study of baboons as a case in point, exhibiting four substantive theoretical and methodological challenges that affect the field more generally: 1) properly characterizing features of the faculty of language in the narrow sense; 2) defining and probing for human language-like computations; 3) distinguishing between linguistic competence as opposed to linguistic performance; and 4) uncovering evidence for evolutionary continuity in linguistic capacity. Our intent is to be critical in the service of clarity, in what we agree is an important approach to understanding how language evolved.

1. Introduction

The last few decades have witnessed renewed interest in the origins, phylogeny, and adaptive significance of human language, in many ways reviving in many ways Lennenberg's (1967) original vision for the field (see, e.g., Pinker and Bloom 1990; Hauser, Chomsky, and Fitch 2002; Jackendoff, 2002). In turn, this has reinvigorated research into relevant comparative data, including evidence from archaeology, paleontology, and animal cognition. Nevertheless, as we illustrate below, considerable confusion remains regarding the central theoretical issues and core concepts to be engaged in this research, leading to empirical studies that are sometimes well off the mark.

Perhaps nowhere has this confusion been greater than in the reaction to issues raised in Hauser, Chomsky, and Fitch (2002), vis-à-vis comparative studies on artificial language learning in animals (see, e.g., Gentner et al. 2006; Abe and Watanabe, 2011; for refutations of these results see Corballis, 2007; Ten Cate, 200x; and Beckers et al., 2012). Here we focus on a recent study in this journal by Rey, Perruchet, and Fagot (2012) as an illustrative case study. We aim not to discourage such comparative work, but rather to sharpen discussion so that the empirical work is aimed at the appropriate conceptual issues that should be of interest to all language researchers.

We focus on four problems in Rey and colleagues' baboon study that continue to hinder investigation into the evolution of language. First, it fails to distinguish between the faculty of language in the broad sense and the faculty of language in the narrow sense (FLN), as outlined in Hauser et al, with a corresponding lack of focus on experimental tests that could detect human-like language competence. In particular, Rey et al. explore the capacity

of baboons to process center-embedded patterns as their experimental test of FLN, and so, they suggest, what is uniquely human about human language syntax. But in fact this approach does not align with Hauser et al.'s approach, which posits *recursion and mappings to the sensory-motor and conceptual-intentional systems* as the core of the human faculty of language in the narrow sense (FLN). Further, Rey et al. appear to equate center-embedding patterns (and sometimes their associated structures) with recursion. This is incorrect. We show that their center-embedding test is both too strong and too weak to serve as an appropriate signal of human linguistic competence. More generally, there seems to be little recognition of the deep divide between generative procedures and recursive functions on the one hand, and finite association-driven computations on the other. Rey et al. position the latter as the scaffolding on which center-embedding – and so human language-like competence – is built; again, we demonstrate that this does not seem to be correct.

A second misunderstanding in Rey et al. centers on the notion of *recursion* itself. Rey et al. define this as "a computational device that calls itself" (180). This definition calls to mind the image of a computational process that itself necessarily forms a self-embedded pattern of postponed computations, e.g., $f(f(\ldots f(*)\ldots))$, which Rey et al. run together with the presumptive center-embedded structures associated with their experimental artificial language patterns. But in fact this image is not at all correct; all such computations can be carried out iteratively instead (see Rice 1965; Soare 1996; Chomsky 2009).

Third, Rey et al.'s methodology probing baboons for human language-like abilities — massive training with reward for correct productions — bears little or no resemblance to experimental child language acquisition research **[JW: references]**, nor to child language development as it is currently understood. Consequently, Rey et al. cannot draw conclusions about either human language developmental trajectories or representational outcomes. There is no evidence that baboons attain the competence with respect to center-embedded patterns that children do; the baboons do not exhibit the unbounded, spontaneous, and rapid generalizations typically observed in language development **[JW: references]**.

Finally, the baboons' acquisition trajectories and attained abilities undermine Rey et al's proposal that working memory in conjunction with the addition of "phonological recoding" could serve as an evolutionary "bridge" between human and non-human primates. What is acquired in the case of humans is not simply "more of the same," but something entirely different: a recursive generative procedure that computes a potentially unbounded set of linguistic expressions.

The following sections take up each of these four main points in detail.

2. Testing for FLN

Rey et al. (180) take the "central claim" of Hauser and colleagues to be this: "the ability to process CE structures is a critical cognitive feature distinguishing human from nonhuman communication." To test this claim, Rey et al.

conducted intensive training experiments with captive baboons in which subjects learned associations between pairs of visual shapes, designed to mirror a pattern mimicking the "nested" or center-embedded syntactic structure underlying a sentence such as *the antelope the lion ate ran like a snail.* Such strings form patterns such as $a_1a_2b_2b_1$. The authors suggest that success in this artificial language setting implies that the baboons, relying only on association and working memory, exhibit "a preference for producing responses consistent with CE structure" (182). They further conclude that this amounts to "an alternative view" to the "central claims" of Hauser et al., "that the ability to process CE structures is a critical cognitive feature distinguishing human from nonhuman communication" (180).

However, Hauser and colleagues never asserted that an ability to produce or process center-embedded patterns distinguishes human from nonhuman primates, so this cannot be one of Hauser et al.'s "central claims." Rather, the claim is that only the faculty of language in the narrow sense is "unique to our species," comprising "the core computational mechanisms of recursion as they appear in narrow syntax and the mappings to the interfaces with [conceptual-intentional and sensory-motor systems]" (1573). Here, the term *recursion* is to be construed as a computable function that enumerates or generates a potentially infinite set of hierarchically structured expressions, directly analogous to the inductive generation of sets (and thus the natural numbers, as we discuss below).

Crucially then, the core computational mechanisms of recursion are *not* taken to be the ability to process center-embedded structures, though these distinct notions are often run together, as we note below. Indeed, in several important respects Rey et al.'s string sequences do not serve as particularly sharp probes for uniquely human linguistic competence.. The reason is that the linguistic patterns that are "easy" and "difficult" for people to process do not align well with center-embedded word sequences and their possible foils – such patterns are both too strong and too weak.

As noted by Rogers and Hauser (2010), while people find language patterns in the form a_nb_n difficult to process, as in *people*$_n$ *left*$_n$ (e.g., *people people people left left left*), their corresponding paraphrase forms *people who were left* (*by people who were left*)$_n$ *left* (e.g., *people who were left by people who were left left*) seem *easier* for people to analyze—even though several authors, including Rey et al., assume that these latter patterns are within reach of nonhuman animal abilities. In fact, as is familiar from work that Rey et al. themselves cite, as well as from the classic studies by Miller and Isard (1964: 293), the processing of center-embedded structures in humans is known to be limited by working memory. But memory by itself is not an ability or competence. As Rey et al. acknowledge, it is simply the workspace within which particular algorithms are executed. In humans, the independent existence of a particular linguistic competence can be demonstrated by varying performance as a function of computational complexity. Performance continues to improve indefinitely if time and access to external memory are increased. In contrast, for baboons, this effect has not been demonstrated. Rey et al conclude that "increasing the levels-of-embedding could be too demanding [...]" (p.g. 182). However, it is misleading to speculate that "[a]lthough the present results indicate that baboons are not qualitatively limited in processing CE structures, their performance

could be limited quantitatively to the processing of one or two embeddings" (182). Rey et al. provide no evidence to indicate that the qualitative limits do not simply reduce to quantitative limits, that is, that an unlimited competence underlies the baboons' limited performance. Finally, as Rogers and Hauser observe, center-embedded $a_n b_n$ patterns correspond to the "simplest" possible kind of embedding structure, allowing only, e.g., Sentences embedded within other Sentences, as in *John knows that the baboon learned language*, but not Sentences embedded within Noun Phrases, as in relative clauses (*the baboon who learned language*), let alone with many other constructions in human language. In short, $a_n b_n$ patterns—Rey and colleagues' proxy for center-embedded structure—are simply not good "human language detectors," being both too simple and too complex. This critique holds independently of the method used to demonstrate how individuals acquire such patterns, a point we explore below.

3. Association-based computation and recursion-based computation

Rey et al. take the central conclusion of their research to "suggest that the production of CE structures in baboons and humans could be the by-product of associative mechanisms and working memory constraints" (183). Putting to one side the point that Rey et al. test whether *human* processing is associative, this statement reveals a misunderstanding of the critical difference between associative mechanisms and the core computational mechanisms that Hauser et al. suggest as the key components for the faculty of language in the narrow sense. This amounts to the distinction between a finite association-based *lookup table* and a potentially unbounded memory computational mechanism such as a *Turing machine,* the latter with its familiar two components, finite control unit that can store instructions and a potentially separate unbounded read/write memory tape. (Alternatively, we could adopt any oe of a number of models that resemble more closely physically realizable computers, such as register machines or stored program computers.) Here Rey et al. apparently lack a clear grasp of the notions of *recursive procedure* or *recursion,* a failing that leads directly to their inadequately formulated experimental probe.

Rey et al. summarize their results as follows: "the [baboon's] preference for producing CE structures requires (1) the capacity to form associations between pairs of elements (e.g., $a_1 b_1$ or $a_2 b_2$) and (2) the ability to segment these associations and maintain in working memory the first element of a pair (a_1) in order to produce later its second associated element (b_1)" (RPF: 182). However, if these are the only constraints required to reproduce the experimental results, they imply that the "language" that must be recognized is strictly finite, in the form $a_i a_j b_j b_i$ for $i,j = 1,\ldots,6$ (with i, j distinct). However, every finite stringset such as this is recognizable by a mechanism that lacks a potentially unbounded read/write memory tape, such as an association-based lookup table, a finite-state machine, or just the finite-state control unit of a Turing machine. As a result, there is no need to posit any embedding structure or underlying grammar whatsoever to process such examples. A grammar-free representation remains as a more parsimonious explanation for the experimental results.

At this point one might object that because a finite-state machine for this task must in effect memorize all possible i,j pairings, the formal result is irrelevant: it requires prior training on pairings such as a_1, a_2, ..., and as Rey et al. point out, the baboons have never been exposed to these pairs. However, there are more compact representations for finite-state languages based on "neuron-like" units that overcome such simple objections. For example, after an

average of 53,349 a_ib_i training repetitions per baboon, it is reasonable to posit that a Hebbian-type "wire-together-fire-together" cell assembly could be in place for each of the six a_1b_1, …, a_6b_6 visual stimuli pairs. In this model, an input stimulus of any a_i or b_i fires off the relevant assembly to expect its matching pair, with an activation strength that diminishes exponentially over time. This suffices to reproduce the patterns the baboons seemingly acquired. If the first two stimuli in a test appeared as the sequence a_1a_2, at times t_1 and t_2, respectively, the stimulus a_1 at time t_1 would initiate an expectation for b_1, with an activation level beginning its decay at t_1. Next, at time t_2, the a_2 stimulus would appear, initiating an expectation for b_2 with an activation level beginning its decay at time t_2. Assuming selection preference to be determined by activation level, then at time t_3, the expected preferred response would clearly be b_2 with its higher activation value; this would be followed by b_1, exactly in line with Rey and colleagues' experimental results. Nothing beyond this kind of simple sequence association is required. Minsky (1967: 36) provides an alternative model in terms of McCulloch-Pitts neurons; a third method is suggested in Section 4 below. While of course we do not know the way in which recognition sequences like these are neurally encoded, the key point is that any number of finite-state models can be used to "memorize" *ab* association sequences without a human –like grammar or an associated hierarchical representation. Rey and colleagues' data cannot distinguish among any of these alternative, and simpler, possibilities. Indeed, since repetitive, conditioned association learning already falls within the abilities that we know nonhuman animals, even invertebrates, possess, this would seem to be the null hypothesis, to be rejected only given strong evidence to the contrary. **[JW: references; Gallistel, 1990, The Organization of Learning, MIT Press]**.

Rey et al. themselves recognize that their demonstration is limited: "A stronger demonstration would certainly require testing baboons on two or more embeddings. However, one may argue that increasing the levels-of-embeddings could be too demanding for baboons" (182). Presumably, to extend the method to include a third sequence pairing would demand extensive additional training. But this in turn implies that the baboon behavioral repertoire lacks the characteristic human language "stimulus free" property. While RPF attribute this to a working memory deficiency (in both baboons and human subjects), or absent "semantics" (in baboons),[1] there remains a crucial difference between baboons and humans. Given external memory aids, the human ability can be extended indefinitely **[JW: references]**. However, in baboons it cannot.

Evidently, Rey et al. fail to appreciate that if one takes the computational theory of the mind/brain seriously, it is the Turing machine (or one of its many equivalents) that serves as the natural model for human abilities, including language. The association-based lookup table model is a nonstarter. As Minsky (1967: 114) puts it, "the extension [to infinite memory] is actually needed to gain any really practical insight into real-life computers. [I]nfinite-machine theory is more realistic than finite theory, for practical purposes [because] it is not the finiteness of the machines that limits their uses." In a similar vein, Gallistel and King (2009: xi) also observe that "The distinction is critical, because [an association-based model, such as connectionism,] tries to dispense with the tape and place all of

[1] Contrary to what Rey et al. imply, much of "semantics" is enabled by, rather than an enabler of, syntactic competence: e.g., semantic representations of predicate-argument structure or discourse-structure are built on syntactic structures.

the memory in the transition table (state memory)" notwithstanding "well-known results in computer science [that] this cannot be a generally satisfactory solution [given] the infinitude of possible experience." Apparently, the distinction between finite and infinite memory, more specifically the independence of assumptions about working memory from those about syntactic competence, has not been ontologically necessary for the bulk of research in human sentence syntax during the past sixty years. Now, it is of course possible that matters could have turned out the other way. In this case, human syntactic competence would depend in some essential way on assumptions about the interaction between human performance, specifically the assumption of a small, finite working memory, and human language competence. (See, e.g., Berwick & Weinberg 1985 for one proposal along these lines; and Chomsky and Miller, 1963 for an earlier proposal.) If interaction were the rule, we would expect to find no law-like regularities regarding syntactic competence that could be stated without reference to assumptions about working memory and the like. But, as the large body of linguistic research attests, this does not seem to be so Rather, by and large it appears that our best accounts of language *in toto* posit two relatively independent theories, one about linguistic knowledge without reference to memory limitations, and one about working memory without reference to linguistic knowledge. When joined together, these two accounts yield the best overall theory of language. (In fact, as established by Chomsky and Miller (1963), and more acutely by Berwick (1982, 1985), this notion of 'best' can be sharpened to mean "smallest" or "most likely" in the minimum description length or Bayesian senses, respectively.)

The same holds for human cognitive abilities generally, and is worth an extended discussion, since evidently the facile "human brain as finite-state machine" viewpoint persists (see Hockett 1955): "any explicit classical cognitive model that takes into account the finite memory resources of the brain is necessarily a finite-state machine (Petersson, Folia, and Hagoort 2012: 91). While it is of course true that the human brain is finite, and so could be represented as a (large) finite-state machine, nevertheless, as Minksy, Gallistel and King, and many others have noted, this is irrelevant, because infinite models for these finite systems demonstrably yield more scientific insight. Consider the case of human arithmetical competence. Here, the cognitive science literature takes for granted that this ability is internalized in the form of rules for addition or multiplication (though perhaps not transparently), with working memory "truncating" this ability in recognizable ways. Neither is there any existential confusion that the same holds for any physically realizable computer like one's laptop. In both cases, the infinite model turns out to yield the right theory.

A typical example can be found in the experimental work of Hitch (1978) on human mental arithmetic. In the case of adding two arbitrary integers to yield their sum, Hitch assumes that there is some step-by-step recursive procedure (an algorithm, a Turing machine) that carries out addition, which he then proceeds to experimentally verify. The algorithm fetches individual digits as provided by some "input-output" system, stores intermediate results with possible carries in working memory, and then manipulates this intermediate result further, accumulating results first in the 1's place, then the 10's place, and then finally the 100's place. Hitch's results confirm this "add-carry-hold" model. To be sure, as Hitch also demonstrates, limits on available working memory impose constraints that lead to predictable performance errors that follow from the "internalized rule" model, crucially even for small

integers. Performance is improved by adding external memory (such as paper and pencil, equivalent to extending the read/write tape), without any need to "reprogram" the internalized algorithm.

What of an association-based lookup table counterpart for two-digit addition, one that would simply memorize all possible additions, like memorizing *a-b* pairings in the case of baboons? One approach would be to store the two integers to add and their resulting sum, so that, e.g., 79+62=151 would be stored as the triple (79, 62, 151). Results would then be retrieved (by association) directly from memory. Suppose then that the table included all the possible one and two digit sums given inputs from 1 to 99. However, this method's behavior would not square with human abilities as described by Hitch. Not only would it fail to reflect the experimentally demonstrated engagement of working memory in even simple one- and two-digit cases, its total failure when extended to 3-digit sums could not be remedied by recourse to additional external memory. The only solution would be an extension of the internal lookup table itself—i.e., the memorization of additional triples, again mirroring the baboon example (and connectionist networks, e.g., Elman 1991). Further, the internalized rule vs. memorized triple approaches differ enormously with respect to the number of examples needed to acquire comparable arithmetic competence. For the internalized rule approach, only the sums for 1 through 9, plus the rules for carries need to be acquired—a fixed amount of information, even though the integers to be added might grow indefinitely large. In fact, as Minsky (1967: 23) shows, if addition is carried out in binary, then a simple two-state, seven-rule finite machine suffices. Note that this kind of rule system makes "infinite use of finite means" in precisely the way claimed for rules of language, discussed below. As far as we know, no one has seriously objected to this notion of "unbounded generative capacity" in the case of addition, even though each novel addition clearly corresponds to a novel behavior. Similarly, there is no "largest" addition problem, just as there is no "largest" syntactic structure. In contrast, the association-table approach requires a ten-fold increase in memory as each new placeholder column is added (e.g., adding any 2 digit integers requires knowledge of about 100 sums; adding any 3 digit integers requires memorizing an additional 1000 sums). With a linear increase in the number of inputs, the number of output combinations grows exponentially, and is thus quite appropriately referred to as a computationally intractable "combinatorial explosion" by Gallistel and King (2009: 97). Further, an association-based approach is not "creative" in the sense of being able to add two integers it has never seen before.

The same rule-based vs. memorized association table distinction holds for multiplication, but with a vengeance: unlike addition, multiplication cannot be carried out by a finite-state machine. The reason is simple, but provides insight into the linguistic analog. As one moves from right to left, from least to most significant digits, accumulated intermediate products can grow arbitrarily large, and so cannot be represented by any device with a finite memory. What is required in this situation is something similar to a Turing machine with a potentially unbounded input/output tape, so that these intermediate results can be written to an external tape and 'carried forward' to later stages of the computation. Any purely association-based method must fail at some point. Yet no one doubts that people have internalized the rules for multiplication and are in principle able to multiply any two integers – just as no one doubts that the same holds for a laptop or even a lowly calculator.

We have dealt at some length with arithmetical competence because it appears to correspond in virtually every respect with that of linguistic competence. As observed above, Turing machines for either arithmetic or language are systems of digital infinity: each enumerates a potentially infinite set of discretely structured objects via computable functions. Indeed, as Chomsky (1959: 137) notes, the grammar for generating a set of linguistic expressions can be characterized as a function mapping the integers onto this set. As hypothesized for the faculty of language in the narrow sense, the discrete elements of a syntactic expression (e.g., words) are read as input and, as instructed by internalized linguistic rules, combined into sets (e.g., phrases) and written on the tape to be carried forward as "intermediate results," serving as inputs to subsequent computations. This enables the potentially unbounded combination of words into phrases, and phrases into sentences, and sentences into discourses.[2]

The correspondence with arithmetic is direct. The generative process just described is essentially the "iterative conception of a set," with sets of discrete objects, linguistic or arithmetic, "recursively generated at each stage" such that "the way sets are inductively generated" is directly analogous to "the way the natural numbers [...] are inductively generated" (Boolos 1971: 223). Language and arithmetic's common unbounded digital character may well not be accidental. Both draw on similar generative procedures, as noted in Hauser *et al*. Though non-human animals appear to be able to carry out some arithmetical operations using analog quantity representations, or perhaps subitizing for small integers, there seems to be no evidence for anything in non-human animals resembling the rule systems sketched above or the generalization to an unbounded domain of arithmetic expressions. And there seems to be no evidence for some "proto-arithmetic" capacity in non-human primates that can be incrementally modified to yield full-fledged human arithmetical competence. Even when animals are given access to the Arabic integers, **[what does this mean? Rcb]** they never acquire anything remotely like the successor function, generalizing beyond the trained input. Moreover, and of direct relevance Rey et al.'s methodology, the research on non-human animal integer processing also demonstrates that it is entirely different from children's development of arithmetical competence: animals never exhibit the kind of inductive leap that all children take once they have acquired knowledge about the first few integers.**[JW probably a reference needed here, from HCF's reference list?]**

Finally, besides a failure to appreciate the divide between association-driven systems and recursive functions, Rey et al.'s account of recursion apparently runs together distinct notions that ought to be kept separate in order to properly formulate experimental probes for human language. At the very outset of their article, Rey et al. state, "recursion is defined as a computational device that calls itself" (180). But if this is meant as a definition of a "recursive function" as used in Hauser et al. (and as formalized by Turing 1936; Post 1936, 1944; Kleene 1952) then this definition is misleading, if not entirely inaccurate. Presumably, this definition is intended to call to mind the notion of an algorithm defined via the use of a recursive procedure that "calls itself." A familiar example is the way that one can define a procedure to compute the factorial of a number, *factorial*(*n*), by means of an algorithmic step that involves multiplying *n* by *factorial*(*n*–1), in this way having the procedure "call itself" But as Rice (1965: 114)

[2] It is over these hierarchically structured expressions that operations like grammatical agreement (e.g., in person, number, gender, case, etc.) and transformations (e.g., "*wh*-movement" forming the interrogative *whom did John see* from the form underlying the declarative *John saw whom*) are defined.

observes, with only one exception to his knowledge,[3] "all functions ever evaluated on computers so far have been *primitive recursive*" (our emphasis), and so are functions that can be implemented without recourse to a "computational device that calls itself" in the sense just illustrated above by the *factorial* procedure. For such functions, a simple iterative process (using "looping") suffices. To demonstrate this, Rice provides an explicit, ten-line Fortran program that can implement any primitive recursive function; famously, Fortran does not provide for recursive procedure calls. Applying this result leads to the well-known iterative procedure for computing *factorial* simply by successively accumulating partial products ($i–1$) x i , adding this to a running total. No recursive procedure calls are required. More strongly still, Rice goes on to note that Kleene (1952) proves a far more general result: the same holds for *all* recursive functions: "there is a general iterative process which applies even to recursively defined functions that are not primitive recursive." Just as with *factorial*, *all* computable functions can be written without recourse to recursive subroutine calls. What it does require is some way to "carry forward" arbitrarily large intermediate results, say by means of an arbitrarily long input/output tape, as in the multiplication and syntactic examples noted earlier.

Why is it important to get the definition of "recursion" correct? First of all, since Hauser et al. hypothesize that recursion is a core component of the faculty of language in the narrow sense, any test of this hypothesis should be clear about the concept. Second, it may be that defining recursion as "computational device that calls itself" leads to the impression that a center-embedding *process* can be taken as the hallmark of recursive computation, and so naturally serve as a litmus test for human-like language, as in Rey *et al.*'s experiments. But this runs together two distinct notions, *computational process* vs. *computational output.* The first pictures how a computational process unfolds in time, while the second specifies what the output must look like. As we have just seen, while the pattern of a computation *f* that "calls itself" in terms of recursive subroutines *may* be written in a 'center-embedded' style, as, e.g., (f (… (f( ) ...))), so that the resulting computational process results in a 'center-embedded' expansion of pending, nested, procedure calls, this is not required. In any case, a procedure's output need not reflect the pattern of its associated computational process – the factorial procedure simply outputs one integer, whether it uses recursive subroutines or not. Whether an output is hierarchically 'nested' or not is a separate question.

.

In short then, the two notions of recursive computation and recursive output should not be conflated. With recursion defined correctly, we can imagine extensions of Rey and colleagues' research that could prove productive. For instance, as discussed in section 2, to sustain the claim that "the limited ability for recursion in animals should [...] be found in working memory limitations [rather] than in an inability to produce recursive structures per se" (Rey et al., p.g., 183), it is necessary to demonstrate a competence that is being limited by performance. Such an experiment would first establish an upper bound on the memory capacity of the organism to process center-embedded structures (presented in an organism-specific modality) and second enable some artificial raising of that upper bound. If performance cannot be enhanced, then it is reasonable to conclude that competence reduces to performance, a result that would be inconsistent with findings for our own species **[JW: references]**.

[3] Rice then proceeds to demonstrate that the one constructive example known of a *non*-primitive recursive function, Ackermann's function, can be readily defined in Fortran, again without the use of recursive subroutine calls.

4. What did the baboons learn?

Rey and colleagues' experiments were designed to show that baboons can learn to produce six visually distinctive *a–b* pairs of shapes, as well as to avoid neutral, "distractor" shapes. Following intensive training to form these associations, the baboons were presented with two test situations designed to probe for presumptively center-embedded patterns, each test consisting of three presentation rounds. In both tests, the baboons were presented with the identical first two rounds: shown an initial $a_i a_j$ sequence (ranging over six possible visual types), the baboons had to select a_i and then a_j, respectively, rather than a visual distractor. In the third round, the baboons were presented with either b_i, b_j, and a distractor (Test 1), or b_i, b_j, and b_k (Test 2, all indices distinct). In the first test condition, baboons consistently selected b_j, b_i, (and not the distractor). In the second test condition, baboons again consistently selected b_j, b_i, in that order (and not b_k). Rey et al. concluded that the "baboons spontaneously ordered their responses in keeping with a recursive, centre-embedding structure" (180).

This conclusion does not follow for at least two critical reasons. First, the training method is problematic. As the authors note, "*To be rewarded* in Test 1, baboon had to touch both b_1 and b_2 *with no constraint on order* and to avoid the distractor. *To be rewarded* in Test 2, they had to touch two of the three displayed shapes *with no constraint on order* (181, our emphasis)." Not rewarding the baboons for *just* the correct order (e.g., b_1, b_2) is correct because otherwise some of the 240 test trials would be deployed as additional training examples. However, , by rewarding the baboons for either order on the third round, they were, in effect, obscuring the "nested" sequence order as the correct one. This is especially problematic given the real possibility of rapid learning of new associations through training in the test conditions.

Second, given the training paradigm, simple association mechanisms can explain the results. For example, consider the case where baboons select a_2 in the second round. In the third round, b_2 is now primed for selection from either the set $\{b_1, b_2,$ distractor$\}$ for Test 1, or the set $\{b_1, b_2, b_3\}$ for Test 2. Then, from the two remaining sets of items, $\{b_1,$ distractor$\}$ (Test 1) or $\{b_1, b_3\}$ (Test 2), b_1 is primed by a_1 from the first round and is accordingly selected. Rey et al. argue that selection of b_1 from the $\{b_1, b_3\}$ set is significant because the baboons were never trained to avoid b_3. However, over the tens of thousands of trials, b_3 had never been associated with a_1, and thus b_1 remains the clear choice. As outlined in Section 3, these results can be explained by any number of purely association-driven models, from simple Hebbian-type networks, to McCullough-Pitts or Kleene "nerve nets," or, to provide one more example, the class of *k*-limited automata—in which "the state of the automaton is determined by the last *k* symbols of the input sequence that it has accepted, for fixed *k*" (Chomsky 1963: 336).

We thus *agree* with Rey et al. that (183): "[the] findings do not imply that baboons possess the innate computational device that has been postulated for humans (i.e., the faculty of language in the narrow sense proposed by Hauser et al., 2002)." This conclusion brings into stark relief, however, that even the authors do not believe that we share with baboons any derived heritage whatsoever with respect to the faculty of language in the narrow sense. Clearly we

take issue with other aspects of their conclusions. From all the evidence, unlike humans baboons do not appear to possess the competence to process arbitrary center-embedded structures, with this ability somehow constrained by their working memory. Rather, the simplest explanation is that the baboons learn sequence pairs by brute-force association, without recourse to sequence "structure." If so, then the baboon results do not provide the sought-for evolutionary bridge between nonhuman and human primates envisioned by Rey et al. The gap between stimulus-bound, association-based computation and the power of laptop computers, Turing machines, or humans looms as large as ever. Beyond center-embedding — which is so confounded by processing factors that inference from behavior to the underlying computation is virtually if not technically impossible — researchers need to devise tests for all aspects of the Turing architecture of the faculty of language in its narrow sense (and related systems, e.g., arithmetic): finitary procedures, read/write memory, recursively generated digital infinity (unboundedness, discreteness), set theoretic output dependencies (nested, crossing, etc.), and the apparent "displacement" of phrases (see note 2), another type of "carrying forward" at the core of human language competence. Devising such tests may prove difficult, but the hypothesis put forward by Hauser et al. is nontrivial once the computational mechanisms of recursion are properly defined and understood.