PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY B

Phil. Trans. R. Soc. B (2012) **367**, 1933–1955
doi:10.1098/rstb.2012.0103

Review

Artificial grammar learning meets formal language theory: an overview

W. Tecumseh Fitch[1,*] and Angela D. Friederici[2]

[1]*Department of Cognitive Biology, University of Vienna, Althanstrasse 14, Vienna 1090, Austria*
[2]*Max Planck Institute for Human Cognitive and Brain Sciences, Stephanstrasse 1a, 04103 Leipzig, Germany*

Formal language theory (FLT), part of the broader mathematical theory of computation, provides a systematic terminology and set of conventions for describing rules and the structures they generate, along with a rich body of discoveries and theorems concerning generative rule systems. Despite its name, FLT is not limited to human language, but is equally applicable to computer programs, music, visual patterns, animal vocalizations, RNA structure and even dance. In the last decade, this theory has been profitably used to frame hypotheses and to design brain imaging and animal-learning experiments, mostly using the 'artificial grammar-learning' paradigm. We offer a brief, non-technical introduction to FLT and then a more detailed analysis of empirical research based on this theory. We suggest that progress has been hampered by a pervasive conflation of distinct issues, including hierarchy, dependency, complexity and recursion. We offer clarifications of several relevant hypotheses and the experimental designs necessary to test them. We finally review the recent brain imaging literature, using formal languages, identifying areas of convergence and outstanding debates. We conclude that FLT has much to offer scientists who are interested in rigorous empirical investigations of human cognition from a neuroscientific and comparative perspective.

Keywords: artificial grammar learning; formal language theory; comparative neuroscience; neurolinguistics

1. INTRODUCTION

Formal language theory (FLT) has its roots in mathematics [1,2] but was established in its modern form by Noam Chomsky in an attempt to systematically investigate the computational basis of human language [3,4]. Since these beginnings, the theory has been continually expanded to cover other scientific domains. The most prominent new application was in computer science, where the study of FLT is now a core part of the standard curriculum, providing the theoretical foundation for fundamental issues such as programming language structure and compiler design [5]. Psychologists have used FLT to explore learning and pattern-processing abilities in humans and other species [6–10], while in neuroscience the theory has been used in neuroimaging experiments to better understand the neural computations of hierarchy and sequence [11,12]. Finally, in biology, FLT has been used to analyse diverse topics such as the structure of RNA molecules [13,14] and the sequential structure of chickadee song [15]. FLT has thus grown far beyond its original roots in language, to become a key component of the theory of computation, applicable to virtually any rule-governed system, in any domain.

In this paper, we review recent progress in applying FLT to empirical research in animal cognition and neuroscience, as well as highlighting some pitfalls that can accompany attempts to merge theory and practice. We start with a non-technical overview of FLT, intended to give an intuitive understanding of the theory and its significance and to provide a gentle preparatory overview for the more rigorous paper by Jaeger & Rogers [16]. We then provide a more detailed analysis of the difficulties involved in translating this body of theory into an empirical research programme. We start with the difficulties caused by the use of infinity as a tool for proofs in mathematics, which leaves such proofs technically irrelevant in the real world of finite brains and finite time. We provide a detailed analysis of one particular rule system, the so-called 'A^nB^n grammar', which has been employed in many recent studies, both in neuroscience and in animal cognition. We suggest that this grammar is appropriate for answering certain interesting questions, but has sometimes been over-extended to address issues for which it is poorly suited, for which we suggest alternative, more appropriate grammars. In the process, we highlight the need to clearly distinguish among a number of separate issues, which—although related—should not be conflated.

* Author for correspondence (tecumseh.fitch@univie.ac.at).

One contribution of 13 to a Theme Issue 'Pattern perception and computational complexity'.

These include notions such as hierarchical structure versus centre-embedding, context-freeness versus long-distance dependency and formal complexity versus recursion. FLT provides the theoretical concepts and terminology to clearly distinguish among all of these terms, and we argue that it should be used to do so more rigorously in the future. We then provide a detailed look at some of the recent brain imaging literature using FLT, highlighting the areas of nascent agreement along with outstanding open questions. We conclude by pointing out some areas within FLT that remain little explored, but might provide fertile ground for future research.

2. FORMAL LANGUAGE THEORY AND THE THEORY OF COMPUTATION

We have an intuitive sense that some cognitive computations are more difficult than others. For most people, it is harder to play chess or solve equations than to buy groceries or drive a car. For most of us, it is more difficult to parse sentences in a non-native language (regardless of our level of proficiency) than in our native language. However, a central finding of computer science is that our intuitions about complexity do not necessarily apply to computer programs. In fact, it has proved relatively easy to create machines that can play chess at a high level, but so far impossible to create adequate car-driving systems. Because of this, an important component of modern computer science is a framework for quantifying the 'difficulty' or 'complexity' of a computational problem or algorithm in terms that are explicit and unambiguous. Starting with the work of the brilliant mathematician Alan Turing, and combined with further insights owing to Gödel, Church, Post, Kleene, Chomsky and many others, FLT has grown today into one key pillar of the theory of computation (and thus compiler design and many other aspects of computer science). The other main pillars are the theory of computability (what problems can or cannot be solved) and the theory of problem complexity (how the difficulty of problems scales with their size) [5,17].

The theory of computation provides the practical basis for software tools we use everyday, which thus provide useful illustrations of the core concepts of FLT. We favour such everyday examples from computers, rather than mathematical formalisms, because we expect that most of our readers will have some experience with the former but not necessarily with the latter. More detailed and mathematical treatments are easy to find [5,17–20], and a paper in this issue provides a particularly accessible formal introduction designed for experimentalists [16].

(a) *Regular expressions*

We start our survey with a simple, well-defined computational system, termed a finite-state machine, which is equivalent to another simple construct called a 'regular expression'. Search functions, such as the *dir* command in DOS, or the *ls* function in UNIX, provide everyday examples. Such functions use a syntax that allows us to search for the arbitrary target pattern in a large database of words and/or numbers. Given a set of file names:

```
> filenames = {a.wav, b.doc, c.bmp, MySong.doc, MySong.wav}
```

running the function:

```
> ls *.wav filenames
```

(or the equivalent with dir in DOS) on this set will return the subset

```
> {a.wav, MySong.wav}.
```

The search string '*.wav' says, in effect, 'give me all the strings that end with ".wav" '. The * character tells the *ls* or *dir* command that the string(s) can start with any characters in the alphabet, we do not care which or how many. This search string is one simple example of a general framework called '*regular expressions*', which provide a very powerful basis for computer-based search that underlies searching, replacing and other functions in many computer programs. This ability to use regular expressions to match patterns was first instantiated in the *grep* function in UNIX/Linux, and has proved so useful that the term 'grep' has entered hacker lingo as a verb meaning 'to search by computer'.

Regular expressions are composed using a few simple but powerful rules and operators, familiar to many computer users. The operator *, as used earlier, means 'any string of any length' and by appending it to our search string (e.g. 'string'), we can find our target pattern even if preceded by anything (*string), followed by anything (string*) or buried in anything (*string*).[1] More specific operators also allow us to specify a single, unspecified character (?), a character from a particular set (e.g. numbers {0–9} versus letters {a–z}), or even a specific number of characters from a certain set. Any time you have some pattern or a set of patterns that can be captured by a regular expression, you can use *grep* to search an arbitrary database for that pattern. You can *grep* for your name or email address or telephone number in the archives of a discussion group, or *grep* for a particular gene sequence in the online human genome database. The search engine Google is an extended version of *grep* that takes the entire web as its database. Regular expressions are at the core of computer search in today's world.

Given this flexibility and power, we might think that regular expressions are capable of specifying any kind of pattern that we can imagine. Crucially, however, this turns out not to be true. For instance, imagine a simple symmetrical pattern where a particular number of items of type A is followed by the same number of a different type B. Examples of this set include {AB, AABB, AAABBB, AAAABBBB, etc.}, and extend indefinitely (so a string of 1346 'A's followed by 1346 'B's is still a member of the set). This pattern is notated A^nB^n in FLT. It is easily proved that this set cannot be specified by a regular expression (see the textbooks listed earlier for mathematical proofs). We conclude from this fact that there are patterns that we can conceive of, and that we could easily (if laboriously) recognize ourselves, but that cannot be captured by a regular expression. Why does this matter? Because, as demonstrated by the mathematician Stephen Kleene

in 1956 in the theorem that bears his name, regular expressions and the corresponding rule sets termed 'regular grammars' are exactly equivalent to one of the most ubiquitous classes of computing devices, which are termed 'finite-state automata' [21].

(**b**) *Finite-state automata*

FLT relies on abstract models of computational systems termed 'automata' (and often, perhaps confusingly, also often called 'machines'). Two canonical examples of such models are the finite-state automaton (FSA) and the Turing machine. Automata such as these are mathematical abstractions, not real devices designed to be manufactured. For example, the Turing machine includes as part of its mathematical definition a storage tape of infinite length, and thus we could never build a real Turing machine. Many automata, although well-defined in theory, are unbuildable in practice (a fact that has some implications that we will discuss later). Despite this, the abstract notion of a Turing machine is extremely important and powerful in mathematics and FLT: infinity is a powerful tool for mathematical abstraction, but not a real thing that we find in the world.

The simplest class of well-defined automata are called finite-state automata because they have a finite number of operating states or 'positions'. The FSA starts at a predefined start state, and then jumps between its other states, depending only on its current input and current state. For each of these jumps, it can emit an output symbol as it hops along. Thus, an FSA can be fully defined by its set of states, its input alphabet (the input symbols that it recognizes), an optional output alphabet (which might or might not be different) and a function that tells it which is the next state to go to, given its current state and current input. Any given FSA is capable of 'recognizing' a certain set of patterns, and rejecting others. By 'recognize' we mean simply that, given this pattern as input, it can generate some particular prespecified output (e.g. 'OK'). The set of patterns recognizable by an FSA may be infinite. Because of this, and its simplicity, the FSA is a good starting point for further discussions of automata and computational complexity. Critically, as already mentioned, Kleene's Theorem demonstrates the equivalence, or interchangeability, of FSAs and regular expressions.

One point, overlooked by many, is that FSAs can recognize (or generate) a simple, long-distance dependency of the start-and-end sort (e.g. ab*a or cd*c) as the automaton in figure 1*c* shows (p. 1103). However, other patterns are clearly beyond the capabilities of an FSA, because we already know that certain patterns, such as A^nB^n, cannot be captured by regular expressions: they are beyond the capabilities of our simplest class of automaton. Thus, something with more computational power is clearly needed.

(**c**) *Turing machines*

Other automata take an FSA as their starting point, and achieve additional computational power by adding some additional form of memory. The most important and powerful such automaton is the Turing machine, which adds to an FSA (the 'controller') a storage tape of unbounded length, on which symbols can be written or erased.[2] Thus, in addition to possessing a large (but finite) set of states that its controller can occupy at any one moment, the Turing machine has by virtue of this tape an additional unlimited form of memory for storing past operations, intermediate results and so on. A Turing machine can easily recognize the A^nB^n language described earlier: it simply stores the number of times A has been repeated (that is, writes successive integers every time its input jumps from A to A) and then compares that with the number of Bs (B to B jumps). Thus, Turing machines are more 'powerful' than their FSA component, in the sense that they can recognize patterns and solve problems unsolvable by any FSA. This is not surprising given their additional resources. What is altogether more remarkable is that the Turing machine is capable of computing ANY deterministic function whatsoever: if something is computable, a Turing machine can compute it. Thus, modern computer scientists accept the Turing machine as their very definition of 'computability', broadly accepting the 'Church/Turing thesis' that a function is computable if and only if it is computable by a Turing machine.

The Turing machine and FSA are well-defined automata that provide useful endpoints for a scale of *computational power*: the FSA provides the lower level (which is powerful and practically useful, but has its limits), while the Turing machine provides the upper limit (it is all-powerful in the sense that, if a function is computable at all, an automaton in this class can compute it). We will use the term 'computational power' in this paper in this specific sense, framed by the specific automata discussed in FLT. We do not mean to imply by this that this is the only way to insightfully characterize the power of algorithms, or that ultimately this is the best way to think

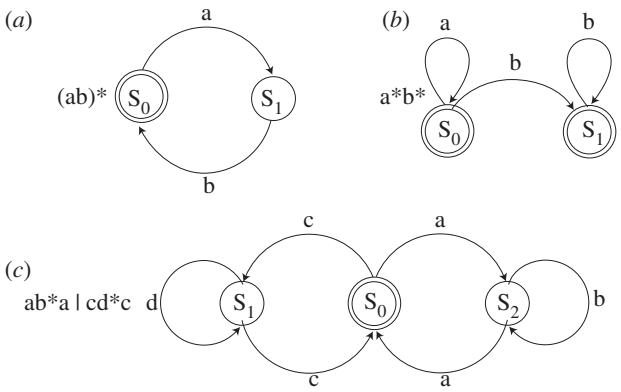

Figure 1. Three examples of simple finite-state automata and their stringsets. Circles represent states, arcs represent transitions between states, with the corresponding symbols, and double circles indicate 'accept' states. (*a*) The (ab)* or $(ab)^n$: accepts strings of zero or more 'ab' bigrams. (*b*) The a*b*: accepts any number of 'a's followed by any number of 'b's. (*c*) A long-distance dependency: this automaton illustrates that FSAs can also check for arbitrarily long-distance dependencies. This grammar accepts strings of the form ab*a, where any number of 'b's can intervene between the two 'dependent' 'a's, (or similarly for cd*c strings).

about the different aspects of brain function. What this specific sense of computational power gives us is an explicit, formal axis along which any particular algorithm can be placed, which thus provides one useful dimension along which to characterize the rule-governed capacities of a machine, or a human or animal subject. Other potentially useful dimensions will be discussed briefly at the end of this paper.

Given these two endpoints, we might immediately ask two questions:

— Are there other intermediate classes of automata, with powers greater than an FSA but less than a Turing machine?
— Where do human computational powers (or those of other species) fall along this spectrum?

(d) *The Chomsky hierarchy*

In an attempt to answer the second question, the young Noam Chomsky and his colleagues built upon the framework already discussed and provided a positive answer to the first question. Chomsky outlined a set of intermediate formal possibilities, between the extremes of Turing machines and finite-state automata, and arranged them in a theoretical hierarchy that now bears his name. Because both the nature and the importance of this hierarchy are sometimes misunderstood, we will try to make clear here both what the Chomsky hierarchy is, and why it is important. First, let us consider the relationship between a Turing machine and an FSA. Because every Turing machine contains within it an FSA, anything computable (e.g. any pattern that can be recognized) by an FSA is perforce computable by a Turing machine. Thus, the set of FSA-recognizable patterns is a proper subset of those computable by a Turing machine. This is obvious from the way in which these automata are defined.

The Chomsky hierarchy incorporates several intermediate levels of automata, which have in common with the Turing machine an additional memory system but discard the assumption that this memory can be freely accessed. For example, a 'pushdown automaton' (PDA) includes an FSA and a pushdown stack (which is a memory that can only return the most recent item placed upon it, like the stack of trays in a cafeteria). Because a stack is more limited than the infinite tape, it is intuitive (and can be shown mathematically) that the PDA is less powerful than a Turing machine, while being more powerful than an FSA. And as before, the set of patterns recognizable by the PDA is a proper subset of those captured by the Turing machine. In fact, a PDA can recognize the A^nB^n language discussed earlier, which is beyond an FSA. Thus, we now have a nested set of patterns, enclosed one within the other like Russian dolls. A second intermediate form of automaton is called a linear-bounded automaton and includes both the FSA and PDA within it. Figure 2 provides an illustration of this nested hierarchy.

The Chomsky hierarchy provides a broad framework for discussing the computational power of automata, universally accepted in this role in theoretical computer

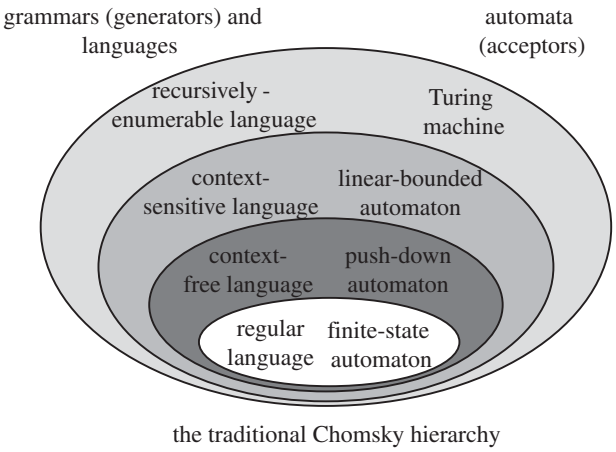

Figure 2. The Chomsky hierarchy for grammars, languages and automata. The Chomsky hierarchy is an inclusion hierarchy that aligns particular forms of grammar, and the languages they generate, with particular classes of automata—abstract computational 'machines' that can be constructed to accept such languages. All of the grey circles beyond the innermost white circle represent the *supra-regular* grammars and languages, which require computational power above the level of a finite-state automaton. See Jäger & Rogers [16] for more details.

science, algorithmic theory, FLT and discrete mathematics. It is well-defined, explicit and unambiguous and discussed and defined in any textbook on these topics. The Chomsky hierarchy contains no hidden assumptions about Universal Grammar, the 'poverty of the stimulus' argument or other of Chomsky's various more controversial ideas about language. A computer scientist using the Chomsky hierarchy need not accept other aspects of Chomsky's thought, any more than a logician using Bertrand Russell's innovations in mathematical logic need accept Russell's pacifism or atheism. While this should be an obvious point, we have been surprised by how often the distinction gets blurred. Although there are other ways to arrange automata in hierarchies of ascending power, as well as finer subdivisions of existing hierarchies [16], we focus on the Chomsky hierarchy in this work owing to its interdisciplinary acceptance and understanding. Anyone who studies basic computer science or the theory of algorithms will be familiar with the framework, and this is more than can be said for any other framework we know of. We do not claim that this whole set of automata (including FSAs and Turing machines) is the best or the most insightful way of arranging the different types of neural computations that we ultimately want to understand as psychologists, biologists or neuroscientists [22]. Indeed, as our understanding of neural computation in vertebrates progresses, it seems likely that different hierarchies will arise, and prove to be more useful. Until such progress is made, however, FLT seems to provide the best theoretical starting point, and indeed has no obvious competition.

One might object that some models of neural computation, especially connectionist networks, offer just such an alternative framework. This is not true. In fact, neural networks are automata, like any other well-defined computational system. Indeed,

surprisingly simple recurrent neural networks can be shown to be Turing-equivalent in principle [23] (although programming such networks to do some of the tasks they could perform in theory has been difficult or impossible in practice). Other classes of automata, such as augmented transition networks, also take their place within this scheme. The power of the Turing machine framework is that it includes *any* computational device: the definition is formally equivalent to a vast number of alternative implementations, in a wide variety of forms (indeed many different computational frameworks were initially offered as alternatives to Turing machines, but were later shown to be Turing-equivalent). Thus, the term 'computation' is used here, and in computer science in general, in a very broad and inclusive sense to capture any algorithm (information processing system). Neural networks and a vast array of other implementations are part of this classification, not alternatives to it.

(e) *Formal language theory*

With these preliminaries behind us, and the basic aims and principles of computational theory clarified, we can now introduce the terminology and principles of FLT. On the one hand, fortunately, this terminology is simple and unambiguous, and thus quite familiar because it maps the core technical concepts onto everyday terms. On the other hand, this familiarity can be deceptive and misleading. As typical with technical terms, we must beware of unwittingly slipping from interpreting the words in their technical sense to their broader everyday sense.

First, the terminology: an *alphabet* A is a set containing a finite number of indivisible symbols. A *string* (often termed, more confusingly, a *word* or a *sentence*) is a finite sequence of symbols, and a *string-over-A* is a finite sequence of symbols from A. The set of all such strings-over-A is denoted A*, pronounced 'A star'.[3] Finally, a *language* over A is any subset of A*. Put verbally, a 'language' in this abstract sense is some set of 'legal' strings from our alphabet A. Put concretely, lets say an alphabet A_1 consists of the digits 0–9. Then any integer number is a member of a language defined over this alphabet (a member of the language denoted by A_1*), but the number 1.35 is not (because the '.' is not in our alphabet). The string 'cat' is, for the same reason, not a member of this language. Similarly, if our alphabet consists of the letters of the Roman alphabet a–z, we could define some more specific languages where 'cat' is contained in the language but 'katze' is not.

These terms and their definitions seem quite intuitive and easily understood. The danger is that they are so easy to understand that it is easy to forget what they do and do not involve. Most prominently, there is no discussion of meaning within this framework. Although we can easily design a grammar, in the technical sense, that can accept 'cat' while rejecting 'katze', this system has no understanding that these strings mean 'feline animal', or indeed that they could mean anything at all. A grammar responds purely to sequence and has no way of embodying meaning or of 'understanding' the signals that are fed to it. These systems are purely syntactic and have no semantics. Thus, 'language' in FLT is missing a substantial component of natural language (the central component of meaning). A 'language' in this formal context is a set of strings (a 'stringset'), and nothing more.

Continuing our exploration of FLT, the simplest way to define a finite language is to simply list all its members. This 'brute force' approach is a possibility as long as the language is finite, e.g. for the set containing the words of English (e.g. a dictionary omitting names, loan words or neologisms). It will not, however, be able to deal with the integers, or the sentences of English, because the number of integers or English sentences is unlimited, and the list could never be complete (any candidate finite set can always be 'trumped' by adding 'Mary thinks that' to a randomly chosen member of the set). If the language in question is infinite, it cannot be listed, and our only hope is to come up with some finite set of rules to generate the language, termed a grammar. A *grammar* is a finite set of rules that specifies some (typically infinite) language. Note that the sentences making up such a set will themselves be finite (in the same way that the set of integers is infinite but each integer is itself finite).[4]

We are finally in a position to restate the Chomsky hierarchy in explicit formal terms. The Chomsky hierarchy incorporates a nested set of automata of increasing power, each of which can generate the strings of some formally defined class of languages. The automata (which have already been discussed) are shown with their corresponding formal language classes, in figure 2. Our old friend, the FSA, is at the centre, with its corresponding language family, the regular languages. Next come the context-free languages, defined by PDAs, which are subsets of context-sensitive languages defined by linear-bounded automata. Finally, in the outer and most powerful ring, we find the languages recognizable by Turing machines. The grammars defining these languages have been given many names, but they were called 'type 0' by Chomsky (who dubbed a certain subset of these grammars 'transformational grammars' [24]).

A subtle terminological difficulty arises from the nested aspect of the rule systems just described. Because FSAs are contained within the class of PDA, the term 'pushdown automaton' *sensu lattu* logically includes all FSAs as a subset or special case. However, these terms are often used informally to exclude their subsets, much as the term 'reptile' is used by biologists to delineate all those descendants of the ancestral reptile who are not birds or mammals (the correct term 'non-avian non-mammalian amniotes' being a bit of a mouthful). Because this type of usage is quite indispensable in the research programme outlined here, we will explicitly define it. Using basic set theory, we can define any of these automata *sensu strictu* in an exclusive manner. Thus, we could use the term 'PDA *sensu strictu*' to delineate all PDAs that are not simply FSAs (in logical terms, this is the set of type 2 grammars omitting the set of type 3 grammars). Of particular importance in the present context are the 'supra-regular grammars *sensu*

strictu', which we define as the class of all automata (that is, all Turing machines) that are not simply FSAs (that is, type 0, omitting type 3 grammars or equivalently 'all grammars above the finite-state level'). This is the sense in which the terms 'supra-regular grammar' or 'supra-regular processing' will be used for the rest of the paper.

(f) *Formal language theory and natural language*

As already suggested, Chomsky's primary interest in formalizing automata and organizing in this way was to provide an initial framework for understanding human natural language ('natural' meaning languages such as Warlpiri, French or English, as opposed to artificial languages such as mathematics, PROLOG or C++). In particular, Chomsky pointed out that English cannot be captured by a regular or 'finite-state' grammar (FSG) because it includes structures (particularly, phrase structures with multiple long-distance dependencies and recursive sentence structures) that are beyond any FSA's capabilities. He further argued that various linguistic phenomena of movement and sentence transformation (e.g. from active to passive) are beyond the capability of context-free languages as well, and thus that natural languages must occupy some broader subset of the type 0 grammars (which he termed 'transformational'). However, it quickly became clear that transformational grammars are in fact too powerful, requiring a cumbersome set of constraints to make grammars with this degree of power tractable. Subsequently, theoretical linguists working within this formal paradigm have gradually honed in on the level of computational power required for natural language [25,26]. After some years of suspicion that context-free grammars were up to the task, it is now clear that certain phenomena of natural language require context-sensitive grammars, and most researchers in this field now agree that human languages require 'mildly context-sensitive' grammars (MSCGs): grammars whose power is just a bit beyond those captureable by a context-free grammar [27–29].

What does it mean to say that 'natural languages require grammars at the mildly context-sensitive' level? First, note that any of these abstract classes of automata, including the weaker class of FSAs, contain many automata far beyond that of any human being. For instance, the Manhattan phone book is a finite list, easily captureable in a simple FSA that has one state for each name/number pair, but this language is far beyond the capacity of any human. Thus, the statement that 'human languages require grammars of at least the power of a finite state automaton' does not imply that the human brain could instantiate *any* FSA. Similarly, just because some animal species can be shown to do various tasks at the finite-state level, we cannot assume that they can induce *any* FSG. Whatever class of computational systems natural language entails, it will always be some subset of the categories of automata described in FLT (figure 2). Thus, in applying the theory to real organisms, we can use it as a general road map, but we never expect any of these very abstract classes of grammars or languages to be co-extensive with our own capabilities (or those of animals or children).

This approach might correctly be termed 'syntacto-centric' [30], but exploiting FLT in empirical work in no way denies the central importance of meaning in language. Rather, it reflects an analytical, 'divide and conquer' strategy that chooses one component of the vast complex of human language, focusing on form rather than content. Fortunately, this form-based approach has been immensely productive in computer science, underlying many of the technological advances we take for granted today, and thus does not seem too limited to be of interest. More importantly, the current understanding of the most complex signalling systems in other animals (for example, bird or whale 'song'), along with other rule-governed systems of humans (e.g. music) suggests that they, like formal languages, are focused on structure and *not* complex meaning encoded into units of the signal. The empirical approach we advocate here relies on explicit formal theories as the basis for experimental design, eschewing questions of signal meaning for the time being. Thus, 'language' in the formal sense used for the remainder of this paper means simply a set of strings defined by some grammar. No notion of meaning is entailed or implied for the 'grammars' we consider: they simply accept or reject strings as belonging to some language.

There are explicit theories that concern the information in signals [31] (although this is quite different from meaning, as emphasized by Shannon [32]), along with semantic models treating meaning in its own right [33,34]. There are also empirical paradigms that focus on the acquisition of meaning by children [35–37] and animals [38]. Finally, it is possible to combine artificial grammar learning with meaning in the laboratory to create 'artificial language learning' experiments [39]. There is thus no conflict between a focus on form (syntax or 'grammar') and content (meaning or 'language')—these are complementary fields of study that must, ultimately, be synergistically combined.

Given this caveat about the purely syntactic nature of FLT, one might ask why anyone should be interested in the question of where human (or animal) capabilities lie in the classification system of FLT. Here are a few reasons, ranging from practical to theoretical. From a purely practical viewpoint, scientists attempting to create computer programs that deal with corpora of data are greatly aided by knowing where the signal-generating system they are studying lies in this system. For instance, the parsing and compiling of either regular or context-free languages are well-defined problems with practical working solutions, but this is not true for context-sensitive languages (*sensu strictu*). The difference between a finite-state and context-free representation is also important to keep in mind when compiling computer code or analysing natural language texts.[5] Knowing whether such texts demand context-free (or higher) powers, or not, is thus very valuable for anyone interested in building fast, robust computing systems. Similarly, recent years have seen an explosion of interest in complex animal signals such as bird song and whale song, and a vast amount of data have been

pouring in from both laboratories and the field that needs to be processed by computer [40]. The tools applied to this task need to be capable, at the formal level, of dealing with the actual complexity of these signals, and if such signals could be shown to require supra-regular grammars, the current crop of finite-state tools typically used to analyse them would be demonstrably inadequate.

In addition to these practical issues, there is a deeper theoretical reason for interest in the formal power of signal-generating systems. The core difference between human natural language and signal-generating systems such as music or birdsong, which can also generate highly complex signals, is that linguistic signals are used to transmit equally complex thoughts from one mind to another. Although music certainly communicates (e.g. mood or emotion, energy and many other powerful and subtle 'messages'), there is no direct correspondence between the units of music (notes, phrases, etc.) and the structure of thought. For all its power and greatness, music simply cannot communicate plebian facts such as 'the lion is in the third cave from the right' or 'you need to soak that nut in water and ashes for two days before you eat it', nor can you use a musical phrase to represent those thoughts to yourself. Indeed, if (as in a few isolated cases) we use musical means such as drums or whistles to communicate thoughts, it ceases to be music and becomes language ('drum talk' or 'whistle languages'). Thus, an intimate correspondence between signal and meaning is the *raison d'etre* of language, and the key factor differentiating it from music, and, as far as we know, the diverse signalling systems of every other species on our planet.

There are many grounds for suggesting that thought itself has a tree-like, hierarchical structure [41]. Research in memory, category formation, word learning, visual cognition, Theory of Mind and many other fields all point to this conclusion [42–44], and Herbert Simon has advanced strong theoretical arguments for why a system of thought *must* have such a structure [45]. But if thoughts have a tree-like structure, and are unlimited in number, any signalling system capable of encoding thoughts (that is, any 'language' worthy of the name, possessing semantics and thus beyond the rigorous confines of FLT) must be able to capture this structure. This is a potentially deep reason that natural languages have, and arguably must have, hierarchical phrase structure (that is, must go beyond simply stringing items together in a simple finite-state system)—they would be inadequate vehicles for thought if they did not. Although, by introducing meaning, this argument clearly goes beyond FLT, it provides the broader context in which these questions become centrally important to anyone interested in natural language as a whole. This has been clear since the beginnings of the discipline: the so-called 'weak generative capacity' (the ability to match stringsets alone) is of quite limited interest. Ultimately, the ability to recover the phrase structure(s) underlying a string (roughly speaking, 'strong generative capacity') is much more interesting, and obviously critical for recovering structured thoughts from linear signal strings. If it could be shown that the signal-generating system of some particular species is limited to a simple serial FSG (e.g. the chickadee calls of Hailman & Ficken [15] and Hailman *et al.* [46]), we need wonder no further why that system is not used to express unlimited combinatoric meanings and complex thoughts. Thus, although FLT only gives us tools for exploring signalling systems as stringset, not as meaningful systems, discovering whether the signals generated and processed in animal communication systems are limited to simple finite-state systems or not will have important ramifications for understanding their capacity to convey meaning, and ultimately for understanding the biology and evolution of language.

3. THE ROLE OF INFINITY IN FORMAL LANGUAGE THEORY

Before we turn to experimental work grounded in FLT, we will briefly discuss the controversial issue of 'infinity' in discussions of language. This is an old issue, nicely encapsulated in Wilhelm von Humboldt's suggestions that human language makes 'infinite use of finite means' [47]. Clearly, every individual human has a finite memory, a finite lifespan and a finite (though astronomical) number of neurons and synapses. This has always been accepted [3,6,48]. Nonetheless, most linguists or computer scientists happily accept that any natural language such as English or Chinese is infinite (in the sense of 'unbounded'), in precisely the same way that the set of the integers is infinite. One argument for this parallels the argument for numbers: if someone claims to have identified the largest possible integer, you can easily prove them wrong by simply adding one to their proposed number. In the same way, any proffered 'longest sentence' x can be trumped by simply generating 'John thinks that x'. Although each of these sentences is of a fixed and a finite length (there are no infinite *sentences*), the *set* of sentences is infinite. From this mathematical perspective, we should no more doubt the infinity of English than we doubt the infinity of the integers.

However, there are more subtle arguments for and against the importance of infinity in natural language [49]. For example, the list of sentences a child hears before fixing on a grammar of English is surely finite, as is the list of all sentences an individual will produce in his/her lifetime. In principle, such lists could be captured by a finite-state system, in the extreme case simply as a list of those sentences. In contrast, all of the proofs used in FLT to demonstrate supra-regularity use the argument of infinity to prove their case (typically this involves invoking the pumping lemma [17]). From a strictly mathematical viewpoint, suspending the axiom that languages are infinite would invalidate most such proofs, and thus greatly weaken FLT.

However, no one supposes that the child simply memorizes all heard sentences: any language user can generate and understand novel sentences, beyond the finite input they received in childhood. *Some* system of more general or abstract rules is necessary to account for this ability. As Chomsky notes [3], 'a grammar must reflect and explain the ability of a speaker to produce and understand new sentences which may be much longer than any he has previously heard'. The minimum that we might

need to account for such generalization is a probabilistic model over a finite-state system (some form or another of a 'Markov process'). But as Chomsky further observes in the same passage 'the point is that there are processes of sentence formation that this elementary model for language is intrinsically incapable of handling', and those include sentences with multiple embedding and nested- or crossed-dependency. Thus, 'the assumption that languages are infinite is made for the purpose of simplifying the description' [3], and to allow mathematical proofs that apply to all and every sentence. But there is no theoretical difficulty at all in limiting stack depth or tape length in supra-regular grammars.

Another convincing counter-argument, owing to Levelt [6], goes as follows. Let us assume that human language use *could* be modelled by a FSA, augmented with transition probabilities (a form of Markov process). For this model to have any psychological validity, a child would need enough data to infer these probabilities from the input. So we can ask what order of Markov approximation would be needed for typical sentences. This reduces to the question 'how many words can separate two words that are dependent upon one another in a sentence?' In the grammatical English sentence 'The woman you recently invited to come to New York and give a lecture in our department seems to be sick', 15 words intervene between the inter-dependent words 'woman' and 'seems'. An attentive English speaker will certainly notice if this pair were incorrectly inflected (e.g. 'The woman ... seem' or 'The women ... seems'). Hence, we would need a k-limited Markov source with $k = 15$ to capture this dependency reliably. But Levelt shows that even with unrealistically lenient assumptions [6, vol. 3, p. 76], such a Markov grammar would require an enormous number of parameters, on the order of 4^{15}, or more than one billion. The busy child would need to set about 30 parameters per second, throughout all of childhood, to assimilate such a model. Thus, a finite-state model must be limited to be learnable (and thus unable to deal with long-distance dependencies) *or* it could be theoretically adequate but, owing to the huge number of parameters, practically useless as a model of the child. This argument can be made from many different perspectives, but will always come to the same conclusion: given realistic assumptions, no regular grammar can adequately model English or any other natural language. This is both a practical and a theoretical conclusion.

This combination of arguments has led most commentators to accept the supra-regular hypothesis for humans. Of course, *any* model of human cognition will make simplifications, and thus will be inadequate in certain ways. This is intrinsic to model-building. What we seek are models that make the right generalizations, and that fail in reasonable ways. For example, we can make a simple modification of the weakest supra-regular system, a push-down automaton, in which the stack memory is of a fixed, limited depth. Such a model has no problem with long-distance dependencies, but it *will* have problems with multiple levels of embedding. This is precisely what is observed in humans experimentally, in abundant psycholinguistic research [50,51]. In other words, supra-regular models with finite stores (limited stack depth or tape length) fail in ways that seem much more realistic as models of human performance.

In conclusion, we should not conclude from the importance of infinity in formal mathematical proofs that infinity plays a central role when we turn to practical empirical issues. Infinity is a powerful tool for abstraction, and its judicious use in mathematics allows a kind of certainty that is wonderfully satisfying. For example, Fourier's Theorem *proves* that any complex signal can be built up by a series of sine and cosine waves. Unfortunately, the proof requires an infinitely periodic signal (which continues unvarying from the infinite past to the infinite future), as well as an infinite set of sine waves. Despite these unrealistic assumptions, the discrete Fourier transform, applied to real signals, turns out to be an incredibly powerful tool at the heart of every mobile telephone and spectrographic programme on the planet today. In the same way, we can readily assume that PDA's stack or our Turing machine's tape will be of limited depth, and try to match this to empirical observations of finite humans. Although in doing so we lose the ethereal certainty of theorems, we lose few if any of the practically relevant insights of FLT.

4. EMPIRICAL INVESTIGATIONS USING FORMAL LANGUAGE THEORY

In the rest of this paper, we explore how FLT can be used, practically, by biologists, psychologists and neuroscientists, to design and execute experiments and analyse the resulting data. Often, such studies use artificial grammar learning (AGL) paradigms, see [52]. Because most of the recent literature reviving the supra-regular hypothesis has focused on the finite-state/context-free distinction, we start with a detailed investigation of one particularly simple supra-regular grammar: the 'counting grammar' A^nB^n, which has been the focus of numerous recent studies.

(a) *A^nB^n: a model supra-regular grammar*

The stringset defined by A^nB^n, in which the number of 'A' units is precisely matched by the number of 'B' units, has played a prominent role in the development of FLT. It is a textbook example of a simple language that cannot be captured by a regular grammar, as already discussed. Despite its ubiquity in the theoretical literature, to our knowledge, the first use of this grammar in experiments was that by Fitch & Hauser [8], who compared the acquisition of two different grammars in two different species: humans and cotton-top tamarins, a New World monkey species. One grammar was the simple regular grammar $(AB)^n$, which entails any number of 'AB' units, and the other was A^nB^n. In both cases, the units were consonant–vowel speech syllables, with the A units spoken by a human female and the B units by a male. Fitch & Hauser found that, while college undergraduates were able to master both grammars, the monkeys only showed above-chance rejection of non-grammatical stimuli for the regular grammar. The monkey's success on the regular grammar showed that the techniques were adequate to

elicit rule learning, with generalization, from this species. Fitch & Hauser concluded from this pair of results that monkeys 'can spontaneously master' the regular grammar, but are unable to cope with the supra-regular grammar, and thus that 'tamarins are unable to process a simple phrase structure' (where 'phrase structure grammar' was explicitly defined to mean a supra-regular grammar *sensu strictu*). This conclusion is clearly consistent with the *supra-regular distinctiveness* hypothesis discussed in Fitch *et al.* [52].

Unfortunately, this conclusion was immediately misinterpreted as concerning 'recursion', in a commentary in the same issue by David Premack, which stated 'In a paper on page 377 of this issue, Fitch & Hauser report that tamarin monkeys are not capable of recursion. Although the monkeys learned a non-recursive grammar, they failed to learn a grammar that is recursive. Humans readily learn both.' [53, p. 318]. This was an unfortunate mischaracterization, because the Fitch & Hauser paper drew no conclusions about, and indeed made no mention of, recursion. Their inference was explicitly focused on the supra-regular boundary, which has no clear relationship to recursion or recursive rules (see below). It was quickly pointed out that there are many ways to recognize the A^nB^n language [54], only some of which might necessarily involve recursion. Unfortunately, the incorrect belief that A^nB^n provides a litmus test for recursion was further perpetuated by a second study testing for recognition of the A^nB^n grammar, this time in starlings. Gentner and co-workers [9] found convincing evidence for recognition of A^nB^n and titled their paper 'Recursive syntactic pattern learning by songbirds' (although in the text of this paper, the authors apparently recognize that the actual property being tested is context-freeness). In a commentary on the starling paper, Gary Marcus [55] stated that 'The A^nB^n language is generally assumed to be recursive'. As a result of these multiple characterizations, there is now considerable confusion in the literature about what, exactly, mastery of A^nB^n (or other supra-regular grammars), by humans or any other species, is supposed to indicate. We now discuss the possibilities.

(b) *Mastery of A^nB^n indicates a supra-regular system*

From the viewpoint of FLT, a system's ability to recognize the stringset generated by A^nB^n tells us one thing, and one thing only: that the system is supra-regular (beyond finite state), and therefore has some form of auxiliary working memory, such as a push-down stack, counter or tape, that is not available to an FSA. Although a weak automaton that can recognize A^nB^n is a PDA, with a stack depth limited to the maximum value of n, any more powerful automaton (such as a linear-bounded automaton or a Turing machine) can also recognize (or generate) this stringset. Thus, simple mastery of this grammar by some system is not sufficient to tell us *where* in the nested class of systems occupying the supra-regular portion of the Chomsky hierarchy it lies: only that it is supra-regular.

This ambiguity has important implications for the parsing of A^nB^n strings, as illustrated in the structural diagrams of figure 3. Each of three diagrams exemplifies

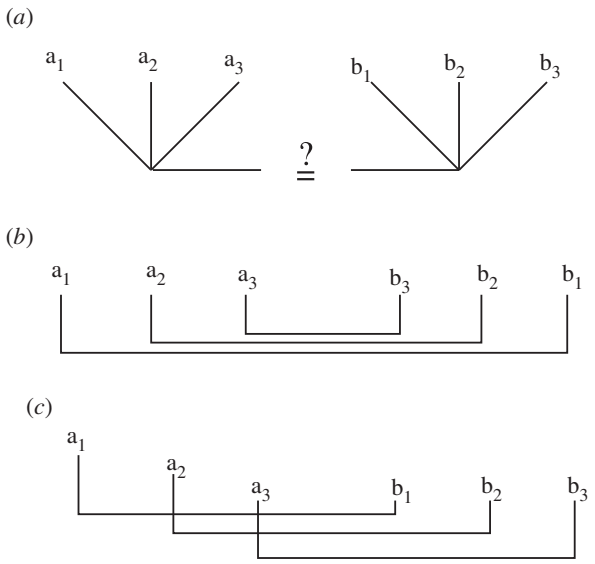

Figure 3. Three possible strategies, and corresponding structures, for recognizing A^nB^n. (*a*) The simplest strategy is 'count-and-compare': count the number of 'a's and the number of 'b's, and then accept the string if they are equal. This strategy is supra-regular, and generates a single hierarchical level. (*b*) An alternative strategy yields a 'nested' or 'centre-embedded' structure, and is a natural strategy for a pushdown automaton because it matches each 'b' with the most recently seen 'a'. (*c*) A third strategy yields a 'crossed' dependency, and cannot be accomplished with a single pushdown stack. It thus requires at least a context-sensitive grammar.

a different computational mechanism able to recognize the A^nB^n language. The top-most, which is the most obvious and appears to capture what humans spontaneously do when confronted with this stringset, could be called 'count and compare'. This involves simply tallying the number of 'A's, storing that number, tallying the number of 'B's that follow and tallying *that* number and then comparing the two. This could be implemented by an integer register that is incremented by one for each A, and decremented for each B (it should then be 0 for grammatical strings). Two registers could also be used, one holding each of the two counts, and comparing them (figure 3*a*). A PDA could recognize the same stringset by storing the number of 'A's as a series of marks, pushed one by one onto a stack, and then 'erased' by removing them from the stack for each corresponding B. An MCSG compatible solution would be to write a 1 for each A on a tape. Once the B phrase starts, this system would rewind, and then cross off a one for each successive B. In either of the last two cases, an empty stack or a blank tape would be required for acceptance. Crucially, *all of these alternative algorithms require supra-regular processing resources* (whether register, stack or tape), and each suggests a different order of processing. What is relevant then is that there is some way of representing the exact number of 'A's, and if this value is not bounded *a priori* to any fixed number, the system is perforce supra-regular. No conclusions about recursion are warranted.

(c) *Eliminating finite-state 'cheats' with generalization and mis-matched foils*

What evidence is needed for us to conclude that a system 'recognizes' the A^nB^n language? As figure 4

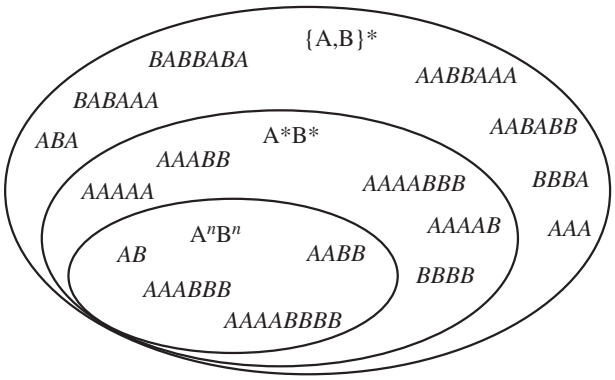

Figure 4. Regular string supersets for A^nB^n. Although recognition of the specific stringset A^nB^n requires a supra-regular grammar, various regular languages contain A^nB^n strings as special cases. For example, the regular language A*B* includes all strings of 'A's followed by 'B's, including those where the number happens to be the same. Similarly, the regular language {A,B}* simply means 'any string of 'A's and 'B's' and also obviously includes A^nB^n strings as a special case. Thus, perhaps non-intuitively, the most inclusive languages (the outer circles of the figure) require less powerful computational machinery.

illustrates, there are many possible regular grammars that could accept strings of this language, and correctly reject many others, and we need to exclude these alternatives if we wish to infer that our system instantiates a supra-regular grammar [9,54]. For example, the set of all strings made up of 'A's and 'B's (written {A,B}*) includes A^nB^n as a subset, as does the set of all strings that start with 'A's and end with 'B's (written A*B*). Similarly, the union of two regular grammars, A^2B^2 and A^3B^3, accepts all A^nB^n strings where $n = 2$ or 3. These and other regular grammars provide potential 'cheats' that would allow a regular system above-chance performance in experiments like these.

While it is difficult to exclude all *possible* regular grammars, we can eliminate most of the reasonably simple ones by employing foils (to exclude the overly general cases) and extensions (to exclude the overly specific grammars). Thus, after exposure to A^nB^n strings where $n = 2$ or 3, we can then test our candidate system with A^4B^4 ($n = 4$) strings. If the system has induced the supra-regular rule, it should accept these generalizations. In contrast, the 'regular union' grammar given earlier would reject such extensions. So this provides one crucial test, allowing us to empirically exclude overly specific regular grammars. Both humans [8] and starlings trained on A^2B^2 [9] accept such generalizations, suggesting that neither species implements overly specific templates to identify their stringsets [56].

A second possibility is an overly lenient grammar that accepts the target strings but many others besides. A particularly crucial superset of A^nB^n is the regular language A*B*. A system implementing this grammar would accept all A^nB^n strings and correctly reject ABAB or BAAB strings. The crucial test in this case is the 'unmatched foil' A^nB^m, where $n \neq m$. Such strings will be accepted by A*B* or similar variant regular grammars, but clearly rejected by any system implementing A^nB^n. Although Fitch & Hauser did not test for this, several later studies [11,57] showed that humans spontaneously reject such unmatched foils, strongly suggesting that they induce the supra-regular grammar as opposed to A*B*. Starlings also rejected such mismatches [9].

However, there is a third and more subtle possibility, noted by van Heijningen *et al.* [58], that different subjects might implement different regular grammars, and that the composite result (if all individuals are lumped together) might appear to constitute significant evidence for supra-regularity, even if the behaviour of each individual subject is consistent with a simpler set of regular rules. Excluding this hypothesis involves either statistical analysis by individual and/or a maximum-likelihood approach, where each grammar is treated as a hypothesis, and the likelihood that this would generate the observed accept/reject data from one or more birds is calculated. Placing zebra finches in an operant set-up very similar to that of Gentner & co-workers [9], these authors [58] argued that both their birds and Gentner's might be 'succeeding' on the task using a motley collection of regular grammars [59]. While it is fair to say that this question remains open, these data opened the door for the most recent study.

Abe & Watanabe [60] used a habituation–dishabituation to probe pattern perception in Bengalese finches (*Lonchura striata domestica*), and again provided evidence for learning of the A^nB^n grammar in this species. In this case, rather than operant training, a mere exposure paradigm was employed, and vocalizations produced to different grammars and their violations were used as a dependent variable. The authors found that listening birds chirped more to novel A^nB^n strings, including novel extensions to $n = 4$. Unfortunately, they do not appear to have tested their finches with 'unmatched foils' A^nB^m, where $n \neq m$, and thus we cannot exclude the regular grammar A*B* based on these data. The reason, presumably, is that the authors were focused on a different question: item-wise dependency in relation to 'centre-embedding'. For further critique of this study, see Beckers *et al.* [61] and ten Cate & Okanoya [59].

(d) *Long-distance dependency versus 'centre-embedding'*

This brings us to a second widespread misconception about the A^nB^n grammar: that it necessarily involves centre-embedded dependency relationships between particular 'a' and 'b' items [62]. There are two reasons that this assumption is incorrect. First, as clarified earlier, although recognizing A^nB^n requires a supra-regular system, we have no basis for assuming any *particular* supra-regular automaton must be used to do so. While one might suggest that a context-free grammar is the most parsimonious assumption, and therefore that nested matching would be most natural (figure 3*b*), a system that possessed a tape (like a linear bounded automaton or Turing machine) might just as well implement a cross-serial matching as in figure 3*c*. This seems particularly likely in the case of humans because we know that cross-serial dependencies are required in some languages such as Dutch or Swiss German (and we can thus infer that humans possess capabilities above context-free, see earlier text). So if the system did infer dependencies between items, there is no compelling reason to assume that these would be nested rather than crossed.

A more important reason is that most grammars capable of recognizing the A^nB^n language make no demands that particular A items should match particular B items. Indeed, one simple way to write this grammar involves a random selection of A and B terminals (figure 3*a*). A different version that would entail dependencies between specific A and B items seems, in principle, more complicated (figure 3*b*). Put in terms of the various mechanisms discussed earlier, there is no reason for an automaton recognizing A^nB^n to write individual 'a's or 'b's to its stack or tape memory: it suffices to simply put any mark (e.g. a 1) for *any* A, and then subsequently count or erase them for each B. It is thus not surprising that humans exposed to A^nB^n strings do not keep track of or notice any particular correspondences, even if the experimenter employed a grammar like figure 3*b* to generate them [62]. Since neither of these two grammars is more correct, it is in no sense a failure if human subjects exposed to strings from figure 3*b* induce the grammar in figure 3*a*, because both are fully adequate grammars for recognizing A^nB^n. The assumption of a centre-embedding item-wise dependency appears to rest on confusion between phrasal dependency (which A^nB^n obviously has) with item-wise dependency (which it does not necessarily have).

Of course, it remains an interesting question what kinds of dependencies humans (or birds) exposed to A^nB^n strings attend to, or can learn, and a considerable literature has grown up exploring this topic, further discussed by several of the papers in this issue. Several commentators have concluded that the 'count and compare' option is not particularly relevant to human language, and so although this potential strategy would be supra-regular, it would be of less interest than the centre-embedded or serially linked options [57,63]. Very briefly, two early studies with humans found that humans exposed to A^nB^n strings generated with item-wise dependencies (as in figure 3*b*) failed to notice these dependencies [57,62]. It is worth noting that Perruchet & Rey [62] employed neither generalization over n nor 'mismatch' foils, and thus the conclusions they can draw from their study about supra-regularity are weak. In contrast, while the Dutch subjects in de Vries *et al.* [57] did successfully reject such unmatched strings, demonstrating their acquisition of a supra-regular rule, they did not recognize violations of centre-embedding dependencies. This led to the provisional conclusion that subjects in these studies had mastered the stringset using 'counting' or some similar strategy, rather than embedding. However, several later studies demonstrate that, given proper training, humans can learn *either* nested or crossed dependencies in an A^nB^n framework [64–67], and Bengalese finches may spontaneously master at least symmetrical centre-embedded dependencies [60]. Thus, all three of the structures in figure 3 can be acquired by human subjects, depending on the conditions.

In summary, humans exposed to A^nB^n stringset spontaneously appear to adopt the simplest strategy—matching the number of 'A's with that of 'B's—rather than inferring item-wise dependencies. However, with adequate training, humans can induce grammars over any of the three possible structures in figure 3. We stress that all of these are supra-regular, and that although several of the earlier-mentioned studies have been framed as critiques, they all confirm the basic capacity of humans to master this stringset. At issue, then, is not the supra-regular hypothesis, but the 'item-wise centre-embedding' hypothesis. Unfortunately, this is not a hypothesis that the A^nB^n grammar is well suited to test: other supra-regular grammars seem much more suited to address this question. In particular, the mirror grammar (written as ww^R, where w represents any string and R indicates 'reversed') is another supra-regular grammar, recognizable by a context-free grammar, well suited to examine pattern-based centre-embedding. A mirror grammar over {A,B} generates strings such as ABBA, BAAB, BABBAB, etc., in which the right half mirrors the left half (and incidentally contains all of A^nB^n as a subset).

(e) *What is the A^nB^n grammar good for?*

Assessing this ongoing debate, it seems reasonable to ask whether a further study of the well-studied A^nB^n language is useful. This of course depends on the questions one is attempting to ask. Those who have employed it with human/animal comparisons have, for the most part, been focused on the 'supra-regular distinctiveness' hypothesis, and for this, the A^nB^n grammar is and remains a valid tool [8,9,11,58]. In contrast, most human-only studies have focused on the 'centre-embedding hypothesis' and drawn negative conclusions about the relevance of this type of grammar for natural language, at least if humans can recognize its strings via the 'count and compare' strategy [57,62,63,68]. The argument in this case is that because natural language does not implement counting of words and comparing across phrases, this computational ability is of little interest in understanding language evolution.

There are two answers to this question. The first is that, if one is focused on the ability to infer grammars beyond the regular or finite-state level, 'count-and-compare' is just as squarely beyond this level as is the mirror grammar. Crucially, a substantial animal cognition literature demonstrates that many vertebrates *can* count, exactly, for small integers up to four or five [69–73]: one reason that all of the animal studies discussed earlier used small phrase sizes, of four or below. But recognizing A^nB^n requires more than simple counting: the system must count and compare across phrases. The evidence from animals, thus far, suggests that *this* computation, unlike counting, is difficult or impossible for most tested non-human species. This failure seems very relevant to any detailed analysis of the computational capabilities of different species' brains.

It is also important to remember that operations that seem intuitively 'simple' to us may not be at all simple to other organisms. A good example of this is the detection of bilateral symmetry, which seems so automatic and trivial to us as humans that it might seem to be a very basic and primitive operation. However, considerable research indicates that, at least in those species tested, a generalized notion of bilateral (or mirror) symmetry is not obvious to animals, and

indeed may be beyond reach, even with training [74,75]. In FLT, mirror symmetry detection is another computational operation requiring at least a context-free grammar. This capability can be probed with the same A^nB^n or mirror grammars that have been used to generate written or spoken stimuli (Stobbe *et al.* [56] provide more evidence of a limitation to sub-regular visual computation, in pigeons and parrots).

If there is a fundamental computational restriction that prevents most species from accessing even bilateral symmetry or 'count-and-compare' strategies, this is surely relevant to these species' inability to acquire the syntax of natural language, which by all accounts require supra-regular capabilities of at least this level of computational power. We certainly encourage the testing of multiple species with many other supra-regular grammars, but as a particularly simple starting point, the A^nB^n grammar seems well suited for testing the 'supra-regular distinctiveness' hypothesis.

A variant of this positive answer is provided by multiple brain imaging studies in humans that suggest that the use of the A^nB^n grammar (even if implemented by 'count-and-compare') activates different neural processing routines from the $(AB)^n$ and similar regular grammars [11,66,76]. We will discuss these findings, which remain contentious, later, but here it suffices to note that the specific regions engaged are strikingly similar to those activated in natural language syntax tasks, whose relevance to human language cannot be questioned [77,78].

This literature also illustrates a potential pitfall of the A^nB^n grammar. From the viewpoint of FLT, the question about whether a species (or a brain region) can cope with supra-regular stringsets needs to be separated from questions of centre-embedding (which is one of several possible strategies for processing A^nB^n) or recursion (which is not a question that can be answered with this type of experiment). While the A^nB^n grammar is simple and well suited for investigating the basic issue of supra-regularity, those interested in issues of dependency might benefit from branching out to other grammar types. For example, the 'mirror grammar' ww^R is not just supra-regular, but its recognition requires long-distance dependencies between classes, and these dependencies are centre-embedded. In contrast, the 'copy grammar' *www* has cross-serial dependencies, which require a supra-context-free computational capacity (and thus a linear 'tape' form of working memory, rather than a push-down stack that can cope with A^nB^n or the mirror grammar). We suggest that pitting the mirror and copy grammars against one another may be more rewarding than continuing to apply A^nB^n to questions it is not the best tool for.

5. VARIATIONS OF TESTING PARADIGMS: A PLETHORA OF CHOICES

One difficulty in comparing results across multiple species, or even across studies with humans, lies in the variation in testing paradigms. Starting with humans, and restricting ourselves to the A^nB^n grammar, stimuli have been presented in the domains of written, spoken or synthesized syllables, have involved explicit training with feedback or virtually no instructions ('mere exposure') and have employed explicit yes/no answers (verbally [62], or via a computer interface [8,11,63]). Other AGL studies have used more exotic stimuli, including musical or tactile inputs [79].

Regarding animal experiments, there is so much variability that few fair comparisons can be made across species. One key difference concerns training and reinforcement. Humans readily acquire multiple grammars, including A^nB^n, without training or reinforcement in 'mere exposure' paradigms using only positive examples. In contrast, most animal studies have involved tens of thousands of reinforced trials over months or years [9,58,80], although a few studies use spontaneous behaviour (e.g. looking behaviour) to investigate what types of 'pattern conception' are used spontaneously, without training [8,60]. Each of these approaches has advantages and shortcomings: for exploring fundamental computational limits, training regimes are superior because a failure after extended training is more convincing. For exploring spontaneous learning or species proclivities, mere exposure and looking time provide more relevant information.

Another important difference between techniques is whether single grammars or dual grammars are presented. In traditional human AGL work, as in spontaneous techniques using animals, positive exemplars from a single grammar are first presented in the exposure phase, and then single-test stimuli are presented to be accepted or rejected [8]. This allows researchers to investigate what is learned from exposure to positive exemplars. In contrast, most training research with feedback requires negative exemplars to be presented as well. So, in a two-alternative forced-choice paradigm, one of the choices must be incorrect and thus stem from some other grammar than the one of interest. Even if the 'positive' exemplars (the ones whose choice gives a reward) are from a particular grammar, subjects might be learning to avoid the second grammar. This complicates the design and interpretation of such experiments.

In general, there are too few studies in which different species are given the same stimuli in comparable tasks to permit fair comparisons. The failure of tamarins in a spontaneous task involving A^nB^n [8] cannot be directly compared with the apparent success of starlings on the same grammar [9], given that the starlings received tens of thousands of trials of training with feedback to achieve this success. The best species comparisons available to date are those that use spontaneous or mere exposure techniques to compare humans and animals [8,81] or which provide training to both species [56]. Comparisons across animal species will require collaboration among laboratories, and clear decisions about the hypotheses to be tested and resultant experimental design.

One final issue of comparability is particularly salient in auditory AGL tasks using vocalizations. Fitch & Hauser [8] used recorded human voices to test both monkeys and humans, and it might be that these stimuli are less salient to tamarins than conspecific vocalizations might be (though the monkeys' success

on the $(AB)^n$ task indicates that they do pay attention to patterns in human voices, which is unsurprising given captive animals' close and dependent relationship to humans for feeding and care). Human voices have also been used successfully with rats [80]. However, most animal studies have used conspecific vocalizations to build up the test strings [9,58,60], giving greater success (but unfortunately have not tested any other species, such as humans, with the same strings). We suggest that in the future, animal researchers test humans with identical strings to allow at least this species comparison. Another way around this problem is to use strings built up of abstract sounds (e.g. synthesized musical sounds), or abstract images [56,82] to allow a fair, neutral comparison among different species.

6. EMPIRICAL INVESTIGATIONS II: NEUROSCIENTIFIC DATA

The second major wave of research capitalizing on the AGL/FLT combination is in neuroscience, and particularly human brain imaging research. An important research paradigm using the FLT framework is the AGL learning paradigm introduced more than 40 years ago by Reber [83]. In these pioneering behavioural studies, Reber used AGL to demonstrate that humans can implicitly learn rule systems consisting of a set of non-linguistic rules governing the concatenation of meaningless letter strings. In this work, FSGs were used as a model rule system, simple enough to learn, but complex enough to be challenging. But the focus in this literature was on the learning and its implicit/explicit nature rather than on the grammar itself, and despite scores of publications this research paradigm apparently never ventured beyond regular grammars [84–86].

In contrast, a new and fast-growing field has used FLT to design many different grammars, including supra-regular grammars, to probe the neural mechanisms that underlie abstract pattern recognition abilities in humans, and compare them with those involved in natural language processing. Such studies involve brain imaging technologies such as electroencephalography (EEG) and magnetic resonance imaging (MRI). EEG measures brain electrical activity, while functional MRI (fMRI) images blood flow. Both techniques thus index cognitive function in the brain. Transcranial magnetic stimulation (TMS) uses a powerful magnetic field to perturb neural function, allowing experimental evaluation of the role of particular brain areas. Finally, diffusion-weighted MRI or diffusion-tensor imaging (DTI) can be used to image both grey matter anatomy and the white matter fibre tracts connecting different brain regions that constitute a network. A crucial aspect of the new brain imaging studies lies in the comparison of the brain activation for artificial and natural grammars to investigate whether particular types of artificial grammar recruit brain regions and networks used in processing natural language grammar.

(a) *Natural language data*

One focus of particularly intensive, and controversial, research in recent neurolinguistics concerns Broca's area, or the left lateral prefrontal cortex more generally, and its role in processing linguistic syntax and sequential patterns. The term 'Broca's area' refers anatomically to the pars opercularis and pars triangularis in the left inferior frontal gyrus (LIFG) and cytoarchitectonically to Brodmann's areas BA 44 and 45 [87,88]. In Broca's original article, this area was seen as a speech production centre, but seminal research from Caramazza, Zurif and co-workers revealed that this same region plays an important role in syntax processing during comprehension as well [89]. This has emerged as an extremely robust finding in the neuroscience of language [90,91], but consensus concerning its exact significance remains elusive [92–99].

The debate concerns the degree to which Broca's area is specialized for syntax processing or even linguistic processing more generally, or rather subserves some domain- and modality-general computations such as hierarchical planning, working memory or selection among competing alternatives. Advocates of different models often have different theoretical backgrounds, and adjudication is made difficult by the theory-specific characterization of each different model. One of the first models [92] assigned the computational role of Broca's area to a particular form of syntactic working memory needed to process syntactically complex sentences, thereby specifically relevant for syntactic processing.

In a related, but more specifically linguistic, model, Grodzinsky [93] suggested a syntax-specific role for Broca's area, based on the traditional generative notion of 'syntactic movement'. 'Movement' refers to a particular syntactic computation that can be described as follows. Complex sentence structures often feature words that make reference to distant 'empty' slots in the same sentence, and a traditional approach to such long-distance dependencies is that they result from an abstract 'movement' of the word away from its original location. Oversimplifying for clarity, if we start with the sentence 'John likes sandwiches', we might construct the interrogative 'What does John like?' by changing 'sandwiches' to 'What' and then moving this word to the front of the sentence. Thus, the interrogative *wh-* word 'what' is linked to the empty slot at the end of the sentence, where the direct object of 'like' would normally go (this is thus termed '*wh-* movement'). Grodzinsky & Santi [100] reviewed data from aphasic patients, suggesting that only sentences that possess this type of linkage are particularly difficult for Broca's aphasics, and thus that the role of Broca's area is best characterized by the computation of 'movement'. Using fMRI, they provided additional evidence in support of this view [101].

A neuroanatomically more fine-grained model suggested by Friederici [94] argues for a functional subdivision of Broca's area into an anterior part (BA 47/45) responsible for semantic relations, and a posterior part (BA 44) subserving processing of syntactic relations, in particular long-distance dependencies involving syntactic transformations [102]. A recent overview [91] of fMRI studies on syntactic complexity in different natural languages and different sentence structures including 'movement', 'scrambling'[6] and 'nesting' revealed a clear involvement of Broca's area as syntactic complexity in these constructions increases.

Across these different studies, 'movement'-related activation is localized in the more anterior-ventral part of Broca's area (BA 45), whereas 'scrambling' clusters in the more posterior part of Broca's area (BA 44). 'Nesting', i.e. the processing of centre-embedded structures, was only investigated in a few fMRI studies. One study that evaluated the processing of centre-embedded sentences found activation in BA 44 increased with the number of embeddings [103]. The three models discussed so far [92–94] all focus on multiple long-distance dependencies, which—as we have seen—typically entail supra-regular processing resources (e.g. a register, stack or tape).

These three models posit an exclusive role for at least some portion of Broca's area in processing syntax. An alternative framework has been suggested by Hagoort [95], who offered a broader characterization of the role of the LIFG as subserving linguistic unification, applying across the domains of phonology, syntax and semantics. 'Unification' is a powerful computational operation in which pairs of structures can be combined repeatedly to form larger structures, if certain matching conditions are met. Linguistic unification typically starts by combining lexical items, which have certain necessary syntactic properties (e.g. some verbs may require both a subject and an object, and the verb phrase formed by unifying all these could later be used in a larger construction). Unification is a core operation in many modern grammatical formalisms (e.g. categorial grammar and tree-adjoining grammar) as well as in programming languages such as Prolog [28,104,105]. Unification, in general, requires supra-regular computational resources [27].

Thus, all of these models have in common that they attempt to single out a certain component of syntactic complexity for which Broca's area plays an important processing role. In each case, sophisticated models have been proposed to ground the neuroscientific results in linguistic theory, but in each case the terms used are specific to a given theoretical framework (even though they can cope with the same syntactic phenomena). Progress in testing these different ideas requires an over-arching framework broad enough to encompass all of these possibilities, and precise enough to specify their differences from one another and from other hypotheses about the computational role of Broca's area. FLT, and the theory of computation more generally, seems well suited to this role. In particular, it is noteworthy that *all* of the earlier-mentioned characterizations of the role of LIFG share the necessity of supra-regular processing resources.

(b) *Artificial grammar-learning data*

Neuroimaging studies on AGL have tried to elucidate the neural mechanisms of language learning and have provided insight into the nature of knowledge that is learnt in artificial grammars of a given type. Different studies have focused on different aspects of rule and grammar learning.

The first generation of AGL neuroimaging studies focused on the neural correlates of implicit leaning using Reber-type FSGs. They revealed that two aspects of learning, namely similarity-based learning and abstract rule learning, were correlated with two different brain systems. The former learning type was seen to involve the left hippocampus, whereas the latter was found to activate anterior prefrontal cortices bilaterally [106]. Other AGL studies reported activation in the anterior part of the middle frontal gyrus and the parietal lobe bilaterally [107]. Although the term 'grammar' was used, these studies focused on the learning of rule-based sequences rather than on language processing. And indeed, the brain activation patterns reported in these studies were somewhat different from the neural network for natural language processing, which usually recruits the left temporal cortex and left inferior frontal cortex [91,108]. Also, event-related brain potential studies reported different patterns for the processing of linguistic and non-linguistic artificial grammars. Violations in non-linguistic sequences (i.e. Reber-type grammars) revealed a domain-general centro-parietally distributed positivity around 300 ms [109], called P300 and considered to be domain-general [110]. In contrast, learners of a linguistic artificial grammar (i.e. mimicking the phrase structure of natural languages) demonstrated an early anterior negativity to violations in the string [111], similar to syntactic violations in natural languages [112–116].

The second generation of neuroimaging studies tried to relate AGL research more to natural language research, testing artificial grammars more similar to natural grammars, for example, a grammar named BROCANTO [111,117,118] (figure 5). BROCANTO is a simple FSG with a restricted number of syntactic rules and a restricted number of words in different syntactic word classes (e.g. nouns, verbs, etc.). Using this type of grammar, an fMRI study found that the initial phase of learning was correlated with high activation in the left hippocampus known to support item memory consolidation, but that the activation in the left hippocampus decreased and activation in the left Broca's area increased as learning progressed. This

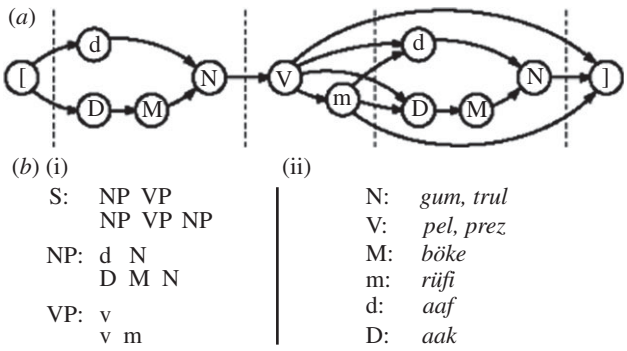

Figure 5. Artificial grammar of BROCANTO. (*a*) Transition from the left to the right following the arrows generates a sentence. For example, {dN Vm DMN}, {dDMN v} and {dN v dN} are correct sentences generated from this automaton. The nodes represent syntactic categories: N (noun), d and D (determiner), v (verb), M (adjective) and m (adverb), and '[' and ']' represents the start and end symbols. (*b*) The rewriting rules (i) of the grammar. The rules define the derivation of S (sentence), NP (noun phrase) and VP (verb phrase) from the terminal symbols given as a set of novel vocabulary (ii).

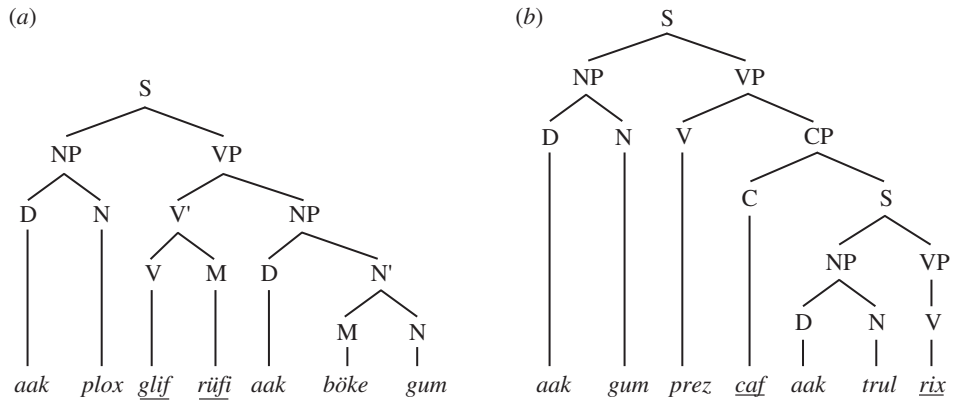

Figure 6. Phrase structures for modified version of BROCANTO. BROCANTO was modified to investigate the difference between grammars that have long-distance dependency (indicated by underlined element (*a*,*b*)) and those that do not. Moreover, the introduction of the complementizer required a word order change in the subordinate clause: from (*a*) verb second position in the main clause to (*b*) verb final position in the subordinate clause. (*a*) Structure with local dependencies. Dependent elements are underlined. (*b*) Structure with long-distance dependencies. Dependent elements are underlined. A set of rewriting rules builds a hierarchical structure. The rewriting rules are represented as binary branches, e.g. S → NP VP. S, sentence; NP, noun phrase; VP, verb phrase; CP, complementizer phrase; D, determiner; A, adjective; N, noun; V, verb; M, verb modifier; C, complementizer.

learning-related change in the brain activity was interpreted to reflect a transition from similarity-based learning supported by the hippocampus to a syntactic rule-based processing in Broca's area [118]. A follow-up study used a variant of BROCANTO (figure 6, introducing a complementizer that obligatorily required a word order change) to investigate local and long-distance dependencies in a sentence. While the processing of a dependency in a local phrase structure activated the left ventral premotor cortex (BA 6), the processing of a long-distance dependency in a hierarchical structure activated Broca's area (BA 44) [119], indicating a difference in the functional neuroanatomy of these two dependency types. Other studies have used 'jabberwocky' sentences to investigate grammatical processing [99,120,121] and found that the network centred on Broca's area was sensitive to grammatical hierarchy, irrespective of whether meaningful or meaningless words were used [99,121].

Another interesting fMRI study compared the learning of a natural and an unnatural language [122]. Italian natives had to learn Japanese or a language with Japanese words but a syntax disobeying the principles of any natural grammar. For the language that followed the universal principles of natural grammars (Japanese), an increase in activation in the left Broca's area and the right prefrontal was found, but not for the language disobeying such rules. This study suggested that artificial grammars following the principles of natural grammars recruit Broca's area (BA 44/45), while artificial grammars disobeying these principles do not.

An early neuroscientific study explicitly adopting an FLT framework, published by Friederici *et al.* [11], contrasted the supra-regular A^nB^n grammar with the regular $(AB)^n$ grammar in a between-subjects fMRI design. They found increased activation of the frontal operculum (an area immediately ventral to Broca's area) for syllable sequences of both grammars, suggesting a role in immediate sequencing. Additional activation of BA 44, in the heart of Broca's area, occurred only with the supra-regular A^nB^n grammar. They concluded by hypothesizing that BA 44 is particularly activated in tasks requiring 'the computation of hierarchical dependencies'. In a further important finding, this study used DTI to examine the white-matter connections stemming from these two different areas, and uncovered two separable neural networks. While the frontal operculum had preferential connections to the anterior temporal lobe via the uncinate fasiculus, BA 44 connected via an independent dorsally located white matter fibre tract to the posterior and middle temporal lobe: the Broca-to-Wernicke connection via the superior longitudinal fasciculus and the arcuate fasiculus. For further discussion of these tracts see later text.

This paper prompted several further studies working with the same grammar in different ways. One critique suggested that, by using a simple A^nB^n grammar with no item-wise dependencies, these experiments were not getting at the linguistically interesting aspect of centre-embedding [57,62]. As discussed earlier, this argument conflates two different issues that FLT neatly separates: recognition of a supra-regular string-set (which can be achieved in various ways, including the supra-regular 'count and compare' strategy), and the structures inferred during string parsing (which might involve simple phrasal chunks, or centre-embedded or cross-serial item dependencies; figure 3). For researchers interested in demonstrating supra-regular computational systems, which of these particular strategies is chosen is irrelevant, because they all go beyond the capabilities of a finite-state system.

Nonetheless, when subjects master the A^nB^n language, it remains interesting to investigate which strategies are used under what circumstances. A considerable literature has now developed that explores this question in detail, often including a brain imaging component [57,62,63,66,68,76,103]. The literature has asked whether, after exposure to A^nB^n strings with predictive dependencies (e.g. $A_1A_2B_2B_1$), subjects notice a violation of this dependency (e.g.

rejecting strings where the A^nB^n rule is obeyed, but the dependency is violated, e.g. $A_1A_2B_1B_2$). After initial results suggesting that 'mere exposure' to centre-embedded dependencies in an A^nB^n grammar is not enough for subjects to recognize violations of those dependencies [57,62,63], it has become clear that human subjects *can* learn such centre-embedding, and extend it correctly, but that this takes additional training, exposure or prosodic structure in the training stage [66,123,124]. Several laboratories have now produced convincing demonstrations that humans can learn both nested and crossed dependencies [64,66]. Intriguingly, processing of such dependencies appears to engage the inferior frontal gyrus, and a TMS study showed that this area plays a causal role in such processing [125].

An fMRI study designed to force the processing of centre-embedded relations (e.g. $A_1A_2B_2B_1$ whereby A_1 and B_1 do not represent particular items, but a class of items) again showed activation of Broca's area (BA 44 in particular) together with motor cortical (SMA) and subcortical areas (basal ganglia) for the processing of A^nB^n grammar sequences compared with $(AB)^n$ sequences [66]. Thus, fMRI studies indicate that both predictive and non-predictive stringset activate Broca's area (BA 44 in particular). Finally, although this literature has primarily focused on centre-embedding, it is also of interest to ask whether humans can learn cross-serial dependencies between items (e.g. with an exposure set involving $A_1A_2B_1B_2$ dependencies [64]). As Uddén & Bahlmann [65] review later in this issue, the answer to this question is positive.

Other recent fMRI studies using AGL further investigated Reber grammars, challenging the view that Broca's area is particularly involved in processing non-adjacent dependencies. These studies reported activation in BA 44/45 for a regular grammar learning and classification task [125–127], but also additional large activations in parietal, occipital and temporal brain regions. The finding that simple right-linear grammars activate Broca's area challenges the view that this area is specifically involved in the processing of non-adjacent, higher order hierarchical dependencies [128]. Rather, on the basis of these latter fMRI studies, Petersson and co-workers suggested that Broca's area is 'a generic on-line structured sequence processor active at different levels depending on the processing complexity' [126,128]. It remains to be resolved how this generic hypothesis can account for the different activations observed in $(AB)^n$ and A^nB^n grammars, given that these two grammars have the same number of very similar rules [22].

Thus, the combined data from the fMRI studies in artificial grammar learning and from non-language domains suggest that Broca's area supports the processing of structured sequences, and of supra-regular sequences in particular.

Although the literature reviewed earlier illustrates the value of FLT in designing neurolinguistic experiments, we need to separate the question of whether a species (or a brain region) can cope with supra-regular stringset, from questions of centre-embedding (which is one of several possible strategies for processing A^nB^n) and recursion (which, for reasons already clarified, is not a question that can be answered with this type of grammar). While the A^nB^n grammar is simple and well suited for investigating the basic issue of supra-regularity, those interested in issues of dependency might benefit from exploring other artificial grammar types including, for example, the 'mirror grammar' ww^R. This not just supra-regular, but its recognition *requires* long-distance dependencies between classes, and these dependencies are centre-embedded. But the other way to approach this issue is to examine particular sentence structures in natural language.

(c) *Brain activation overlaps in natural language and artificial grammar learning*

With respect to the identification of the brain basis of supra-regular stringset processing and the possible underlying processing strategies, a direct comparison between AGL and natural language processing may be useful.

Such studies shed light on possible strategies underlying the processing of A^nB^n in artificial grammar by directly comparing it with the processing of centre-embedded structures in natural language. In natural language, the respective long-distance centre-embedded dependencies are not dependencies in a symmetrical structure (as in A^nB^n [11]), but in an asymmetrical structure (i.e. the relation between subject noun phrase and the verb [103]). In the case of natural language processing, a multi-layered hierarchical dependency structure must be computed to achieve understanding, whereas for the simple artificial A^nB^n structure this is not necessary [57,62]. Thus, the observed overlap in the activation in Broca's area for A^nB^n in AGL [11] and in natural grammar processing [103] suggests that humans build up structural hierarchies (even if unnecessary) when dealing with artificial A^nB^n structures.

One additional issue needs to be considered when comparing AGL and natural language. In the AGL paradigm, *novel* rules must be learned, whereas linguistic studies examine pre-existing rules from native language processing. In most AGL studies, performance is only 70–80% correct, indicating that the learned rules are not well established. Performance by native speakers in natural language experiments is usually much higher. The performance in AGL experiments rather is more comparable to the performance level to language learners, be it in first language (L1) or second language (L2) acquisition. The brain activation pattern observed for L1 learners [129] and L2 learners [130] who are not yet proficient usually involves not only Broca's area, but large portions of the entire left prefrontal cortex even for the processing of local dependencies (i.e. violations in a prepositional phrase).

This broad activation pattern bears some similarity to the activation reported for some recent AGL experiments [78,127], where in addition to Broca's area, large portions of the prefrontal cortex (BA 6,8,9,46,47) including the frontal operculum and the anterior insula, and regions in the parietal, temporal and occipital cortices were activated (when contrasted against a low-level baseline). One important factor may be that in implicit AGL paradigms, participants are confronted with a novel judgement task immediately before entering the scanner, whether a grammaticality

judgement [78] or preference judgement [77]. This may lead to attention-induced control processes and to an activation of large swathes of frontal cortex during classification, even for a simple Reber-type grammar. This argument accords with the view that prefrontal cortex, including Broca's area, is recruited for processes requiring a high degree of cognitive control [131]. Such controlled processes come into play during L2 processing [130] and during language acquisition [129]. In contrast, they are not necessarily activated during highly trained, automatic processes used in processing native language in the adult brain, or for highly trained processes during AGL. A clear separation in the brain activation for adjacent versus long-distance hierarchical dependencies may thus only be observable in a fully established, mature system.

(d) *Diffusion-tensor imaging: a new perspective*

As already mentioned, one study used an MRI technique that allows one to make images of white matter fibre tracts [11]. In this study, probabilistic fibre tracking was used, placing starting 'seed' points located in the centre of the functional activation for the two grammars (frontal operculum and BA 44, for regular and supra-regular grammars, respectively). The analysis revealed two different fibre tracts: a ventral system connecting the frontal operculum to the anterior temporal cortex via a ventrally located fibre tract, and a dorsal system connecting BA 44 to the posterior temporal cortex via a dorsally located fibre connection, including the superior longitudinal fasciculus and the arcuate fasciculus. These data were taken to suggest two different neural networks, with the dorsal network supporting the processing of hierarchically structured sequences (for a similar functional view [132]).

Despite a long history of associating the arcuate fasciculus with language [133,134], an ongoing dispute concerns the precise function of the dorsal pathway [135–137]. A number of researchers proposed that the dorsal pathway supports sensory-to-motor mapping [108,138], and provided good evidence supporting this conclusion. This controversy may be resolved in the light of a recent finding that indicates that in the adult brain, two *different* dorsal pathways can be distinguished: one connecting the temporal cortex and the premotor cortex (possibly supporting sensory-to-motor mapping) and one connecting the temporal cortex and BA 44 (possibly involved syntactic processing) [139,140] (figure 7).

With respect to the ventral pathway, we may also have to consider two functionally separable fibre tracts. There is large agreement that the ventral pathway connecting BA 45/47 to the middle temporal cortex via the extreme capsule fibre system supports semantic processes [108,132,138,141], but it is still open to what extent a ventrally located fibre tract connecting the frontal operculum to the temporal cortex supports the processing of regular structures [11,142]. Unfortunately, fibre tract results can inform us only indirectly about their potential function as they provide only structural information. Connectivity studies do, however, invite a more serious consideration of neural circuits rather than isolated brain regions [141].

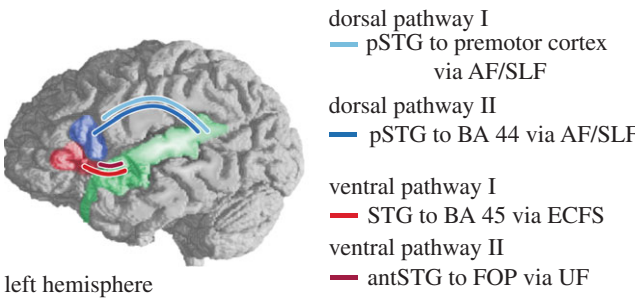

Figure 7. Structural connectivities between the language cortices. Schematic of two dorsal pathways and two ventral pathways. Dorsal pathway I connects the superior temporal gyrus (STG) to the premotor cortex via the arcuate fasciculus (AF) and the superior longitudinal fasciculus (SLF). Dorsal pathway II connects the STG to BA 44 via the AF/SLF. Ventral pathway I connects BA 45 and the temporal cortex via the extreme capsule fibre system (ECFS). Ventral pathway II connects the frontal operculum (FOP) and the anterior temporal STG/STS via the uncinate fasciculus (UF). Reproduced from Friederici [91].

(e) *Evolution and development of brain connectivity*

A useful perspective on the human connectivity data for regular and supra-regular grammars may be provided by adopting phylogenetic and ontogenetic viewpoints. Phylogenetically, recent DTI data show that non-human primates differ from humans in their connectivity pattern. Comparing macaques, chimpanzees and humans, Rilling *et al.* [143] found that the dorsal fibre tract, from Broca's area or its homologue to temporal cortex, gains in strength from macaques to chimpanzees to humans. This difference seems to be of particular interest in the context of the behavioural finding that humans, but not tamarin monkeys, are able to acquire an A^nB^n grammar [8]. This raises this possibility that a robust dorsal fibre tract connecting BA 44 to the temporal lobe is necessary for supra-regular processing, whether artificial or natural.

A complementary perspective is provided by studies of human populations that have problems with the processing of syntactically complex structures in natural language, such as patients with brain lesions [140], or children who still have not reached adult-like performance [144]. Children who still have considerable problems in processing object-first sentences, i.e. sentences in which the object noun phrase is moved to the front of the sentence, have a significantly weaker dorsal fibre tract than adults, while their ventral fibre system is equally strong [144]. This supports the hypothesis that the dorsal tract is functionally relevant for the processing of syntactic hierarchies. Moreover, it is interesting to note that this particular fibre tract (BA 44 to temporal cortex) is not yet myelinated (and thus not propagating information efficiently) at birth, whereas the ventral system and the dorsal tract from the premotor to the temporal cortex is already myelinated at birth [139].

If the functional interpretation of this latter fibre tract as part of the auditory-to-motor mapping system is correct, we would predict that very young infants should be able to recognize simple regularities in the auditory input. There is good evidence that this

is indeed the case, for both AGL [145] and natural language [146]. So far, however, it is not clear to what extent this ability is solely based on the dorsal auditory-to-motor pathway or partly also relies on the ventral pathway present at birth.

7. THE FUTURE

Although the marriage of AGL and FLT has already produced some new and important results, in both the animal and neural domains, this research programme remains in its preliminary stages. Current empirical research provides a detailed exploration of only two particular grammars: the complicated regular grammar introduced by Reber [83] and since explored by scores of researchers, and the supra-regular A^nB^n grammar introduced by Fitch & Hauser [8] and since explored by at least six different research groups. However, there are many other grammars, and empirical approaches, worthy of exploration, and we welcome this ongoing broadening of the field. Investigations of FSGs such as edge grammars [147,148] and other subregular grammars [16] may provide a more detailed dissection of computational primitives particularly relevant to animal researchers [59]. For humans, as discussed already, detailed exploration of other simple context-free grammars such as the mirror grammar or copy grammar will probe the limits of our own pattern-discovery abilities. Innovative experimental designs will play an important role in this—for example, using serial-reaction time tasks and cross-modal auditory/visual AGL [67].

Two further directions immediately beckon. FLT has developed considerably in the last decades in two important directions, both of which offer rich opportunities for empirical research in the 'Grammarama' tradition [52]. The first are tree grammars and tree automata, which have been an area of important recent progress in theory [18,149,150] and are beginning to be used as practical models in psycholinguistics [104,151,152]. The tree-adjoining automata of Aravind Joshi require grammars at the mildly context-sensitive level, and thus are computationally equivalent to other linguistic theories such as combinatory categorical grammar, or minimalist grammars, that converge at this level [27,29]. Nonetheless, tree grammars can lead to a rather different view of the nature of computational primitives from those provided by the traditional string set approach, potentially closer to biological and cognitive reality [153].

The second major advance in FLT concerns probabilistic models of syntax [154,155]. In such models, the apparatus of FLT (typically finite-state or context-free grammars) is augmented by calculating a probability for each production or subtree [16]. Although symbolic models and probabilistic or statistical models are often contrasted by cognitive scientists, there is a growing realization that there is no conflict between these approaches (indeed, traditional rule-based approaches are just a special case of probabilistic models, but where the probabilities are either 0 or 1). Owing to the need for large amounts of data to calculate the required probabilities, such models initially found powerful application in the domain of corpus linguistics [156]. Applying such models to experimental data will require large samples as well, and thus is appropriate for analysing the results of operant AGL experiments [58]. As web-based experiments become more widespread [157], we can anticipate a flood of data from human participants, testing many different grammars with thousands of subjects, that would be well suited to fitting by statistical models.

In the longer term, we can anticipate that this research programme will be able to replace the highly stylized models of computation used in FLT with more biologically grounded computational primitives [158]. While we strongly support moves in this direction, it is crucial that they be grounded in empirical data rather than in intuitions or assumptions about what computational operations are or are not 'primitive'. For instance, it seems intuitive to humans that symmetry recognition should be a very simple and basic operation, and thus part of the conceptual toolkit of most visually-sophisticated organisms. Years of pigeon research demonstrates that this assumption is incorrect, and that generalized bilateral symmetry is in fact an extremely difficult concept for these animals to attain [74]. A similar point can be made about the difficulty of cross-serial versus nested dependencies in human experiments: while most people's robust intuition is that the former should be much easier, what few experimental data is available paint a murkier story.

FLT both allows us to characterize such experiments in more general terms (e.g. that generalized symmetry requires a supra-regular grammar) and provides an explicit language for notating and reporting different string types or grammars (as in $(AB)^n$ or A^nB^n). FLT also, and more controversially, provides a set of 'fixed points' of computational complexity, such as push-down stacks versus the endless tape of Turing machines, that represent abstract representations of different types of working memory. While future research may show that such abstractions are too artificial to usefully characterize neural computational primitives, performing experiments using them will be the surest and fastest way to find out. Currently available alternative models of neural or 'natural' computation [159–162] offer nothing like the scope and specificity, nor the broad acceptance across scientific disciplines, of the theory of computation as treated by FLT.

Thus, we conclude that the empirical research programme combining FLT with AGL will continue, and will continue to be exciting and controversial. Such research will be particularly valuable in uncovering relevant differences among species, and in helping us to characterize the neural mechanisms underlying these differences. This research programme can help itself to a well-developed pre-existing mathematical/computational framework and a rich body of formal understanding and important theorems. If such research eventually leads to a replacement of current formalisms by a computational theory more firmly grounded in biological and neuroscientific reality, it and thus paves the way for its own demise, it will be a welcome sign of scientific progress.

We thank Gesche Westphal-Fitch, Peter Hagoort, Gerhard Jäger, Karl-Magnus Petersson and Nina Stobbe for critical

reviews of earlier drafts of this manuscript. WTF was supported by ERC Advanced Grant SOMACCA and FWF grant W1234-G17. AF was supported by ERC Advanced grant NEUROSYNTAX.

ENDNOTES

[1]The equivalent of the '*' character of *ls* or *dir* (which borrows its symbols from an earlier regular expression system called *glob*) is actually '.' in grep and '*' has a different meaning.

[2]Because any alphabet can be encoded in binary, we can think of these symbols as being limited to ones and zeros without loss of generality.

[3]In formal language theory, the '*' operator, known as a Kleene star, means 'repeat the preceding symbol zero or more times', and thus A^* means 'repeat any of the symbols from the alphabet A zero or more times'.

[4]Note that the number of possible grammars is infinite, but because the number of rules is by definition finite, it is countably infinite (mapping on to the integers). Unfortunately, the number of possible languages is uncountably infinite (mapping onto the real numbers), and this means that there are many possible languages that cannot be captured by a grammar. This might seem like bad news until we realize that all of these 'uncapturable languages' are in some sense trivial assemblages, simple lists that follow no rules. If the language in question obeys any rules, then it can be captured by some grammar.

[5]A good example is provided by the UNIX compiler-building tools, *lex* and *yacc*, which act as a symbiotic pair: *lex* is a FSA capable of parsing regular expressions, typically used to build a lexicalizer or tokenizer to pull out predefined 'words' from a stream of text; *yacc* ('Yet Another Compiler-Compiler') is a tool to build context-free parsers. Because of their higher computational power, *yacc* parsers can easily do things that *lex* parsers cannot (e.g. parse if–then statements, nested loops and recursive parenthetical statements, etc). The price paid is in speed: although *yaac* could in principle tokenize, it would be a hopelessly slow way to do so, while *lex* tokenizers are guaranteed to operate at very nearly the optimal speed possible. Thus, the two tools live in happy symbiosis—perhaps this provides a suitable metaphor for some cognitive subsystems as well?

[6]'Scrambling' refers to noun phrase displacement in the middle field of the sentence in languages with free word order, such as German or Japanese.

REFERENCES

1 Turing, A. M. 1936–1937 On computable numbers, with an application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.* **42**, 230–265.

2 Post, E. L. 1944 Recursively enumerable sets of positive integers and their decision problems. *Bull. Am. Math. Soc.* **50**, 284–316. (doi:10.1090/S0002-9904-1944-08111-1)

3 Chomsky, N. 1956 Three models for the description of language. *IRE Trans. Inf. Theory* **IT-2**, 113–124. (doi:10.1109/TIT.1956.1056813)

4 Chomsky, N. 1959 On certain formal properties of grammars. *Inf. Control* **2**, 137–167. (doi:10.1016/S0019-9958(59)90362-6)

5 Hopcroft, J. E., Motwani, R. & Ullman, J. D. 2000 *Introduction to automata theory, languages and computation*, 2nd edn. Reading, MA: Addison-Wesley.

6 Levelt, W. J. M. 1974 *Formal grammars in linguistics and psycholinguistics: volume 1: an introduction to the theory of formal languages and automata, volume 2: applications in linguistic theory, volume 3: psycholinguistic applications*. The Hague, The Netherlands: Mouton.

7 Pinker, S. 1979 Formal models of language learning. *Cognition* **7**, 217–283. (doi:10.1016/0010-0277(79)90001-5)

8 Fitch, W. T. & Hauser, M. D. 2004 Computational constraints on syntactic processing in a nonhuman primate. *Science* **303**, 377–380. (doi:10.1126/science.1089401)

9 Gentner, T. Q., Fenn, K. M., Margoliash, D. & Nusbaum, H. C. 2006 Recursive syntactic pattern learning by songbirds. *Nature* **440**, 1204–1207. (doi:10.1038/nature04675)

10 Opacic, T., Stevens, C. & Tillmann, B. 2009 Unspoken knowledge: implicit learning of structured human dance movement. *J. Exp. Psychol. Learn. Mem. Cogn.* **35**, 1570–1577. (doi:10.1037/a0017244)

11 Friederici, A. D., Bahlmann, J., Heim, S., Schubotz, R. I. & Anwander, A. 2006 The brain differentiates human and non-human grammars: functional localization and structural connectivity. *Proc. Natl Acad. Sci. USA* **103**, 2458–2463. (doi:10.1073/pnas.0509389103)

12 Pulvermüller, F. 2010 Brain embodiment of syntax and grammar: discrete combinatorial mechanisms spelt out in neuronal circuits. *Brain Lang.* **112**, 167–179. (doi:10.1016/j.bandl.2009.08.002)

13 Knudsen, B. & Hein, J. 1999 RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics* **15**, 446–454. (doi:10.1093/bioinformatics/15.6.446)

14 Sakakibara, Y., Brown, M., Hughey, R., Mian, I. S., Sjölander, K., Underwood, R. C. & Haussler, D. 1994 Stochastic context-free grammars for tRNA modeling. *Nucleic Acids Res.* **22**, 5112–5120. (doi:10.1093/nar/22.23.5112)

15 Hailman, J. P. & Ficken, M. S. 1987 Combinatorial animal communication with computable syntax: chick-a-dee calling qualifies as 'language' by structural linguistics. *Anim. Behav.* **34**, 1899–1901. (doi:10.1016/S0003-3472(86)80279-2)

16 Jäger, G. & Rogers, J. 2012 Formal language theory: refining the Chomsky hierarchy. *Phil. Trans. R. Soc. B* **367**, 1956–1970. (doi:10.1098/rstb.2012.0077)

17 Sipser, M. 1997 *Introduction to the theory of computation*. Boston, MA: PWS Publishing.

18 Levelt, W. J. M. 2008 *Formal grammars in linguistics and psycholinguistics*. Amsterdam, The Netherlands: John Benjamins.

19 Gersting, J. L. 1999 *Mathematical structures for computer science*, 4th edn. New York, NY: W H Freeman.

20 Linz, P. 2001 *An introduction to formal languages and automata*. Sudbury, MA: Jones & Bartlett.

21 Kleene, S. C. 1956 Representation of events in nerve nets and finite automata. In *Automata studies* (eds C. E. Shannon & J. J. McCarthy), pp. 3–40. Princeton, NJ: Princeton University Press.

22 Petersson, K. M. & Hagoort, P. 2012 The neurobiology of syntax: beyond string sets. *Phil. Trans. R. Soc. B* **367**, 1971–1983. (doi:10.1098/rstb.2012.0101)

23 Siegelmann, H. T. 1999 *Neural networks and analog computation: beyond the Turing limit*. Basel, Switzerland: Birkhäuser.

24 Chomsky, N. 1957 *Syntactic structures*. The Hague, The Netherlands: Mouton.

25 Pullum, G. K. & Gazdar, G. 1982 Natural languages and context-free languages. *Linguist. Philos.* **4**, 471–504. (doi:10.1007/BF00360802)

26 Schieber, S. M. 1985 Evidence against the context-freeness of natural language. *Linguist. Philos.* **8**, 333–343. (doi:10.1007/BF00630917)

27 Joshi, A. K., Vijay-Shanker, K. & Weir, D. J. 1991 The convergence of mildly context-sensitive formalisms. In *Processing of linguistic structure* (eds P. Sells, S. M. Shieber & T. Wasow), pp. 31–81. Cambridge, MA: MIT Press.

28 Steedman, M. J. 2000 *The syntactic process*. Cambridge, MA: MIT Press.

29 Stabler, E. P. 2004 Varieties of crossing dependencies: structure dependence and mild context sensitivity.

Cogn. Sci. **28**, 699–720. (doi:10.1207/s15516709cog2805_4)
30 Jackendoff, R. 2002 *Foundations of language*. New York, NY: Oxford University Press.
31 Shannon, C. E. & Weaver, W. 1949 *The mathematical theory of communication*. Urbana, IL: University of Illinois.
32 Shannon, C. E. 1956 The Bandwagon. *IEEE Trans. Inf. Theory* **IT-2**, 3. (doi:10.1109/TIT.1956.1056774)
33 Montague, R. 1974 *Formal philosophy: selected papers of Richard Montague*. New Haven, CT: Yale University Press.
34 Portner, P. H. 2005 *What is meaning: fundamentals of formal semantics*. Oxford, UK: Blackwell.
35 Carey, S. & Bartlett, E. 1978 Acquiring a single new word. *Pap. Rep. Child Lang. Dev.* **15**, 17–29.
36 Bloom, P. 2000 *How children learn the meanings of words*. Cambridge, MA: MIT Press.
37 Hespos, S. J. & Spelke, E. S. 2004 Conceptual precursors to language. *Nature* **430**, 453–456. (doi:10.1038/nature02634)
38 Kaminski, J., Call, J. & Fischer, J. 2004 Word learning in a domestic dog: evidence for 'fast mapping'. *Science* **304**, 1682–1683. (doi:10.1126/science.1097859)
39 Morgan, J. L. & Newport, E. L. 1981 The role of constituent structure in the induction of an artificial language. *J. Verbal Learn. Verbal Behav.* **20**, 67–85. (doi:10.1016/S0022-5371(81)90312-1)
40 Tchernichovski, O., Mitra, P. P., Lints, T. & Nottebohm, F. 2001 Dynamics of the vocal imitation process: how a zebra finch learns its song. *Science* **291**, 2564–2569. (doi:10.1126/science.1058522)
41 Simon, H. A. 1962 The architecture of complexity. *Proc. Am. Phil. Soc.* **106**, 467–482.
42 Miller, G. A. 1956 The magical number seven plus or minus two: some limits on our capacity for processing information. *Psychol. Rev.* **63**, 81–97. (doi:10.1037/h0043158)
43 de Villiers, J. G. & Pyers, J. E. 2002 Complements to cognition: a longitudinal study of the relationship between complex syntax and false-belief-understanding. *Cogn. Dev.* **17**, 1037–1060. (doi:10.1016/S0885-2014(02)00073-4)
44 Happé, F. G. E. 1995 The role of age and verbal ability in the theory of mind task. *Perform. Subj. Autism Child Dev.* **66**, 843–855.
45 Simon, H. A. 1974 How big is a chunk? *Science* **183**, 482–488. (doi:10.1126/science.183.4124.482)
46 Hailman, J. P., Ficken, M. S. & Ficken, R. W. 1985 The 'chick-a-dee' calls of *Parus atricapillus*: a recombinant system of animal communication compared with written English. *Semiotica* **56**, 191–224. (doi:10.1515/semi.1985.56.3-4.191)
47 von Humboldt, W. 1836 *Über die Verschiedenheit des menschlichen Sprachbaues*. Berlin, Germany: Royal Academy of Science Press.
48 Miller, G. A. 1967 Project Grammarama. In *Psychology of communication* (ed. G. A. Miller). New York, NY: Basic Books.
49 Petersson, K. M. 2005 On the relevance of the neurobiological analogue of the finite state architecture. *Neurocomputing* **65–66**, 825–832. (doi:10.1016/j.neucom.2004.10.108)
50 Blumenthal, A. L. 1966 Observations with self-embedded sentences. *Psychon. Sci.* **6**, 453–454.
51 Fodor, J. A. & Garret, M. 1967 Some syntactic determinants of sentential complexity. *Percept. Psychophys.* **2**, 289–296. (doi:10.3758/BF03211044)
52 Fitch, W. T., Friederici, A. D. & Hagoort, P. 2012 Pattern perception and computational complexity: introduction to the special issue. *Phil. Trans. R. Soc. B* **367**, 1925–1932. (doi:10.1098/rstb.2012.0099)
53 Premack, D. 2004 Is language the key to human intelligence? *Science* **303**, 318–320. (doi:10.1126/science.1093993)
54 O'Donnell, T. J., Hauser, M. D. & Fitch, W. T. 2005 Using mathematical models of language experimentally. *Trends Cogn. Sci.* **9**, 284–289. (doi:10.1016/j.tics.2005.04.011)
55 Marcus, G. 2006 Startling starlings. *Nature* **440**, 1204–1207. (doi:10.1038/nature04675)
56 Stobbe, N., Westphal-Fitch, G., Aust, U. & Fitch, W. T. 2012 Visual artificial grammar learning: comparative research on humans, kea (*Nestor notabilis*) and pigeons (*Columba livia*). *Phil. Trans. R. Soc. B* **367**, 1995–2006. (doi:10.1098/rstb.2012.0096)
57 de Vries, M. H., Monaghan, P., Knecht, S. & Zwitserlood, P. 2008 Syntactic structure and artificial grammar learning: the learnability of embedded hierarchical structures. *Cognition* **107**, 763–774. (doi:10.1016/j.cognition.2007.09.002)
58 van Heijningen, C. A. A., de Vissera, J., Zuidema, W. & ten Cate, C. 2009 Simple rules can explain discrimination of putative recursive syntactic structures by a songbird species. *Proc. Natl Acad. Sci. USA* **106**, 20 538–20 543. (doi:10.1073/pnas.0908113106)
59 ten Cate, C. & Okanoya, K. 2012 Revisiting the syntactic abilities of non-human animals: natural vocalizations and artificial grammar learning. *Phil. Trans. R. Soc. B* **367**, 1984–1994. (doi:10.1098/rstb.2012.0055)
60 Abe, K. & Watanabe, D. 2011 Songbirds possess the spontaneous ability to discriminate syntactic rules. *Nat. Neurosci.* **14**, 1067–1074. (doi:10.1038/nn.2869)
61 Beckers, G. J. L., Bolhuis, J. J., Okanoya, K. & Berwick, R. C. 2012 Birdsong neurolinguistics: songbird context-free grammar claim is premature. *NeuroReport* **23**, 139–145. (doi:10.1097/WNR.0b013e32834f1765)
62 Perruchet, P. & Rey, A. 2005 Does the mastery of center-embedded linguistic structures distinguish humans from nonhuman primates? *Psychon. Bull. Rev.* **12**, 307–313. (doi:10.3758/BF03196377)
63 Hochmann, J.-R., Azadpour, M. & Mehler, J. 2008 Do humans really learn A^nB^n artificial grammars from exemplars? *Cogn. Sci.* **32**, 1021–1036. (doi:10.1080/03640210801897849)
64 Uddén, J., Ingvar, M., Hagoort, P. & Petersson, K. M. In press. Implicit acquisition of grammars with crossed and nested non-adjacent dependencies: investigating the push-down stack model. *Cogn. Sci*. (doi:10.1080/03640210801897849)
65 Uddén, J. & Bahlmann, J. 2012 A rostro-caudal gradient of structured sequence processing in the left inferior frontal gyrus. *Phil. Trans. R. Soc. B* **367**, 2023–2032. (doi:10.1098/rstb.2012.0009)
66 Bahlmann, J., Schubotz, R. I. & Friederici, A. D. 2008 Hierarchical artificial grammar processing engages Broca's area. *Neuroimage* **42**, 525–534. (doi:10.1016/j.neuroimage.2008.04.249)
67 de Vries, M. H., Petersson, K. M., Geukes, S., Zwitserlood, P. & Christiansen, M. H. 2012 Processing multiple non-adjacent dependencies: evidence from sequence learning. *Phil. Trans. R. Soc. B* **367**, 2065–2076. (doi:10.1098/rstb.2011.0414)
68 Corballis, M. C. 2007 On phrase structure and brain responses: a comment on Bahlmann, Gunter, and Friederici (2006). *J. Cogn. Neurosci.* **19**, 1581–1583. (doi:10.1162/jocn.2007.19.10.1581)
69 Koehler, O. 1951 The ability of birds to 'count'. *Bull. Anim. Behav.* **9**, 41–45.
70 Boysen, S. T. 1997 Representation of quantities by apes. *Adv. Study Behav.* **26**, 435–462. (doi:10.1016/S0065-3454(08)60385-X)

71 Dehaene, S. 1997 *The number sense*. Oxford, UK: Oxford University Press.
72 Gallistel, C. R. 1990 *The organization of learning*. Cambridge, MA: MIT Press.
73 Uller, C., Hauser, M. D. & Carey, S. 2001 Spontaneous representation of number in cotton-top tamarins (*Saguinus oedipus*). *J. Comp. Psychol.* **115**, 248–257. (doi:10.1037/0735-7036.115.3.248)
74 Huber, L., Aust, U., Michelbach, G., Ölzant, S., Loidolt, M. & Nowotny, R. 1999 Limits of symmetry conceptualization in pigeons. *Q. J. Exp. Psychol.* **52**, 351–379.
75 Swaddle, J. P. & Ruff, D. A. 2004 Starlings have difficulty in detecting dot symmetry: implications for studying fluctuating asymmetry. *Behavior* **141**, 29–40. (doi:10.1163/156853904772746583)
76 Bahlmann, J., Gunter, T. C. & Friederici, A. D. 2006 Hierarchical and linear sequence processing: an electrophysiological exploration of two different grammar types. *J. Cogn. Neurosci.* **18**, 1829–1842. (doi:10.1162/jocn.2006.18.11.1829)
77 Folia, V., Forkstam, C., Ingvar, M., Hagoort, P. & Petersson, K. M. 2011 Implicit artificial syntax processing: genes, preference, and bounded recursion. *Biolinguistics* **5**, 105–132.
78 Petersson, K. M., Folia, V. & Hagoort, P. 2010 What artificial grammar learning reveals about the neurobiology of syntax. *Brain Lang.* **120**, 83–95. (doi:10.1016/j.bandl.2010.08.003)
79 Conway, C. M. & Christiansen, M. H. 2005 Modality-constrained statistical learning of tactile, visual, and auditory sequences. *J. Exp. Psychol. Learn. Mem. Cogn.* **31**, 24–39. (doi:10.1037/0278-7393.31.1.24)
80 Toro, J. M. & Trobalón, J. B. 2005 Statistical computations over a speech stream in a rodent. *Percept. Psychophys.* **67**, 867–875. (doi:10.3758/BF03193539)
81 Saffran, J., Hauser, M. D., Seibel, R., Kapfhamer, J., Tsao, F. & Cushman, F. 2008 Grammatical pattern learning by human infants and cotton-top tamarin monkeys. *Cognition* **107**, 479–500. (doi:10.1016/j.cognition.2007.10.010)
82 Westphal-Fitch, G., Huber, L., Gomez, J. C. & Fitch, W. T. 2012 Production and perception rules underlying visual patterns: effects of symmetry and hierarchy. *Phil. Trans. R. Soc. B* **367**, 2007–2022. (doi:10.1098/rstb.2012.0098)
83 Reber, A. S. 1967 Implicit learning of artificial grammars. *J. Verbal Learn. Verbal Behav.* **6**, 855–863. (doi:10.1016/S0022-5371(67)80149-X)
84 Knowlton, B. J. & Squire, L. R. 1994 The information acquired during artificial grammar learning. *J. Exp. Psychol. Learn. Mem. Cogn.* **20**, 79–91. (doi:10.1037/0278-7393.20.1.79)
85 Knowlton, B. J. & Squire, L. R. 1996 Artificial grammar learning depends on implicit acquisition of both abstract and exemplar-specific information. *J. Exp. Psychol. Learn. Mem. Cogn.* **22**, 169–181. (doi:10.1037/0278-7393.22.1.169)
86 Pothos, E. M. 2007 Theories of artificial grammar learning. *Psychol. Bull.* **133**, 227–244. (doi:10.1037/0033-2909.133.2.227)
87 Amunts, K., Schleicher, A., Bürgel, U., Mohlberg, H., Uylings, H. B. M. & Zilles, K. 1999 Broca's region revisited: cytoarchitecture and intersubject variability. *J. Comp. Neurol.* **412**, 319–341. (doi:10.1002/(SICI)1096-9861(19990920)412:2<319::AID-CNE10>3.0.CO;2-7)
88 Brodmann, K. 1909 Beiträge zur histologischen Lokalisation der Grosshirnrinde. VI, Die Cortexgliederung des Menschen. *J. Psychol. Neurol.* **10**, 231–246.
89 Caramazza, A. & Zurif, E. B. 1976 Dissociation of algorithmic and heuristic processes in language comprehension: evidence from aphasia. *Brain Lang.* **3**, 572–582. (doi:10.1016/0093-934X(76)90048-1)
90 Bookheimer, S. 2002 Functional MRI of language: new approaches to understanding the cortical organization of semantic processing. *Annu. Rev. Neurosci.* **25**, 151–188. (doi:10.1146/annurev.neuro.25.112701.142946)
91 Friederici, A. D. 2011 The brain basis of language processing: from structure to function. *Physiol. Rev.* **91**, 1357–1392. (doi:10.1152/physrev.00006.2011)
92 Caplan, D. & Waters, G. S. 1999 Verbal working memory and sentence comprehension. *Behav. Brain Sci.* **22**, 77–126.
93 Grodzinsky, Y. 2000 The neurology of syntax: language use without Broca's area. *Behav. Brain Sci.* **23**, 1–71. (doi:10.1017/S0140525X00002399)
94 Friederici, A. D. 2002 Towards a neural basis of auditory sentence processing. *Trends Cogn. Sci.* **6**, 78–84. (doi:10.1016/S1364-6613(00)01839-8)
95 Hagoort, P. 2005 Broca's complex as the unification space for language. In *Twenty-first century psycholinguistics: four cornerstones* (ed. A. Cutler), pp. 157–72. London, UK: Lawrence Erlbaum.
96 Thompson-Schill, S. L. 2005 Dissecting the language organ: a new look at the role of Broca's area in language processing. In *Twenty-first century psycholinguistics: four cornerstones* (ed. A. Cutler), pp. 173–190. London, UK: Lawrence Erlbaum.
97 Fedorenko, E., Hsieh, P. J., Nieto-Castanon, A., Whitfield-Gabrieli, S. & Kanwisher, N. 2010 New method for fMRI investigations of language: defining ROIs functionally in individual subjects. *J. Neurophys.* **104**, 1177–1194. (doi:10.1152/jn.00032.2010)
98 Hickok, G. 2009 The functional neuroanatomy of language. *Phys. Life Rev.* **6**, 121–143. (doi:10.1016/j.plrev.2009.06.001)
99 Pallier, C., Devauchelle, A.-D. & Dehaene, S. 2011 Cortical representation of the constituent structure of sentences. *Proc. Natl Acad. Sci. USA* **108**, 2522–2527. (doi:10.1073/pnas.1018711108)
100 Grodzinsky, Y. & Santi, A. 2008 The battle for Broca's region. *Trends Cogn. Sci.* **12**, 474–480. (doi:10.1016/j.tics.2008.09.001)
101 Santi, A. & Grodzinsky, Y. 2007 Working memory and syntax interact in Broca's area. *Neuroimage* **37**, 8–17. (doi:10.1016/j.neuroimage.2007.04.047)
102 Friederici, A. D. 2002 Processing local transitions versus long-distance syntactic hierarchies. *Trends Cogn. Sci.* **8**, 245–247. (doi:10.1016/j.tics.2004.04.013)
103 Makuuchi, M., Bahlmann, J., Anwander, A. & Friederici, A. D. 2009 Segregating the core computational faculty of human language from working memory. *Proc. Natl Acad. Sci. USA* **106**, 8362–8367. (doi:10.1073/pnas.0810928106)
104 Vosse, T. & Kempen, G. 2000 Syntactic structure assembly in human parsing: a computational model based on competitive inhibition and a lexicalist grammar. *Cognition* **75**, 105–143. (doi:10.1016/S0010-0277(00)00063-9)
105 Jackendoff, R. 2011 What is the human language faculty? Two views. *Language* **87**, 586–624. (doi:10.1353/lan.2011.0063)
106 Strange, B. A., Henson, R. N. A., Friston, K. J. & Dolan, R. J. 2001 Anterior prefrontal cortex mediates rule learning in humans. *Cereb. Cortex* **11**, 1040–1046. (doi:10.1093/cercor/11.11.1040)
107 Seger, C. A., Desmond, J. E., Glover, G. H. & Gabrieli, J. D. E. 2000 Functional magnetic resonance imaging evidence for right-hemisphere involvement in processing

unusual semantic relationships. *Neuropsychology* **14**, 361–369. (doi:10.1037/0894-4105.14.3.361)
108 Hickok, G. & Poeppel, D. 2007 The cortical organization of speech processing. *Nat. Rev. Neurosci.* **8**, 393–402. (doi:10.1038/nrn2113)
109 Baldwin, K. B. & Kutas, M. 1997 An ERP analysis of implicit structured sequence learning. *Psychophysiology* **34**, 74–86. (doi:10.1111/j.1469-8986.1997.tb02418.x)
110 Donchin, E. & Coles, M. G. H. 1988 Is the P300 component a manifestation of cognitive updating? *Behav. Brain Sci.* **11**, 357–427. (doi:10.1017/S0140525X00058027)
111 Friederici, A. D., Steinhauer, K. & Pfeifer, E. 2002 Brain signatures of artificial language processing: evidence challenging the critical period hypothesis. *Proc. Natl Acad. Sci. USA.* **99**, 529–534. (doi:10.1073/pnas.012611199)
112 Friederici, A. D., Pfeifer, E. & Hahne, A. 1993 Event-related brain potentials during natural speech processing: effects of semantic, morphological and syntactic violations. *Cogn. Brain Res.* **1**, 183–192. (doi:10.1016/0926-6410(93)90026-2)
113 Hahne, A. & Friederici, A. D. 2002 Differential task effects on semantic and syntactic processes as revealed by ERPs. *Cogn. Brain Res.* **13**, 339–356. (doi:10.1016/S0926-6410(01)00127-6)
114 Isel, F., Hahne, A., Maess, B. & Friederici, A. D. 2007 Neurodynamics of sentence interpretation: ERP evidence from French. *Biol. Psychol.* **74**, 337–346. (doi:10.1016/j.biopsycho.2006.09.003)
115 Kubota, M., Ferrari, P. & Roberts, T. P. L. 2003 Magnetoencephalography detection of early syntactic processing in humans: comparison between L1 speakers and L2 learners of English. *Neurosci. Lett.* **353**, 107–110. (doi:10.1016/j.neulet.2003.09.019)
116 Neville, H. J., Nicol, J. L., Barss, A., Forster, K. I. & Garrett, M. F. 1991 Syntactically based sentence processing classes—evidence from event-related brain potentials. *J. Cogn. Neurosci.* **3**, 151–165. (doi:10.1162/jocn.1991.3.2.151)
117 Opitz, B. & Friederici, A. D. 2003 Interactions of the hippocampal system and the prefrontal cortex in learning language-like rules. *Neuroimage* **19**, 1730–1737. (doi:10.1016/S1053-8119(03)00170-8)
118 Opitz, B. & Friederici, A. D. 2004 Brain correlates of language learning: the neuronal dissociation of rule-based versus similarity-based learning. *J. Neurosci.* **24**, 8436–8440. (doi:10.1523/JNEUROSCI.2220-04.2004)
119 Opitz, B. & Friederici, A. D. 2007 Neural basis of processing sequential and hierarchical syntactic structures. *Hum. Brain Mapp.* **28**, 585–592. (doi:10.1002/hbm.20287)
120 Friederici, A. D., Opitz, B. & von Cramon, D. Y. 2000 Segregating semantic and syntactic aspects of processing in the human brain: an fMRI investigation of different word types. *Cereb. Cortex* **10**, 698–705. (doi:10.1093/cercor/10.7.698)
121 Röder, B., Stock, O., Neville, H. J., Bien, S. & Rösler, F. 2002 Brain activation modulated by the comprehension of normal and pseudo-word sentences of different processing demands: a functional magnetic resonance imaging study. *NeuroImage* **15**, 1003–1014. (doi:10.1006/nimg.2001.1026)
122 Musso, M., Moro, A., Glauche, V., Rijntjes, M., Reichenbach, J., Buchel, C. & Weiller, C. 2003 Broca's area and the language instinct. *Nat. Neurosci.* **6**, 774–781. (doi:10.1038/nn1077)
123 Bahlmann, J., Schubotz, R. I., Mueller, J. L., Koester, D. & Friederici, A. D. 2009 Neural circuits of hierarchical visuo-spatial sequence processing. *Brain Res.* **1298**, 161–170. (doi:10.1016/j.brainres.2009.08.017)
124 Mueller, J. L., Bahlmann, J. & Friederici, A. D. 2010 Learnability of embedded syntactic structures depends on Prosodic Cues. *Cogn. Sci.* **34**, 338–349. (doi:10.1111/j.1551-6709.2009.01093.x)
125 Uddén, J., Folia, V., Forkstam, C., Ingvar, M., Fernandez, G., Overeem, S., Van Elswijk, G., Hagoort, P. & Petersson, K. M. 2008 The inferior frontal cortex in artificial syntax processing: An rTMS study. *Brain Res.* **1224**, 68–79. (doi:10.1016/j.brainres.2008.05.070)
126 Petersson, K. M., Forkstam, C. & Ingvar, M. 2004 Artificial syntactic violations activate Broca's region. *Cogn. Sci.* **28**, 383–407. (doi:10.1016/j.cogsci.2003.12.003)
127 Forkstam, C., Hagoort, P., Fernandez, G., Ingvar, M. & Petersson, K. M. 2006 Neural correlates of artificial syntactic structure classification. *Neuroimage* **32**, 956–967. (doi:10.1016/j.neuroimage.2006.03.057)
128 de Vries, M. H., Christiansen, M. H. & Petersson, K. M. 2011 Learning recursion: multiple nested and crossed dependencies. *Biolinguistics* **5**, 10–35.
129 Brauer, J. & Friederici, A. D. 2007 Functional neural networks of semantic and syntactic processes in the developing brain. *J. Cogn. Neurosci.* **19**, 1609–1623. (doi:10.1162/jocn.2007.19.10.1609)
130 Rüschemeyer, S.-A., Fiebach, C. J., Kempe, V. & Friederici, A. D. 2005 Processing lexical semantic and syntactic information in first and second language: fMRI evidence from German and Russian. *Hum. Brain Mapp.* **25**, 266–286. (doi:10.1002/hbm.20098)
131 Thompson-Schill, S. L., Ramscar, M. & Chrysikou, E. G. 2009 Cognition without control: when a little frontal lobe goes a long way. *Curr. Dir. Psychol. Sci.* **18**, 259–263. (doi:10.1111/j.1467-8721.2009.01648.x)
132 Hoen, M., Pachot-Clouard, M., Segebarth, C. & Dominey, P. F. 2006 When Broca experiences the Janus syndrome: an ER-fMRI study comparing sentence comprehension and cognitive sequence processing. *Cortex* **42**, 605–623. (doi:10.1016/S0010-9452(08)70398-8)
133 Catani, M. & Mesulam, M. 2008 The arcuate fasciculus and the disconnection theme in language and aphasia: history and current state. *Cortex* **44**, 953–961. (doi:10.1016/j.cortex.2008.04.002)
134 Geschwind, N. 1970 The organization of language and the brain. *Science* **170**, 940–944. (doi:10.1126/science.170.3961.940)
135 Friederici, A. D. 2009 Pathways to language: fiber tracts in the human brain. *Trends Cogn. Sci.* **13**, 175–181. (doi:10.1016/j.tics.2009.01.001)
136 Friederici, A. D. 2009 Allocating function to fiber tracts: facing its indirectness. *Trends Cogn. Sci.* **13**, 370–371. (doi:10.1016/j.tics.2009.06.006)
137 Weiller, C., Musso, M., Rijntjes, M. & Saur, D. 2009 Please don't underestimate the ventral pathway in language. *Trends Cogn. Sci.* **13**, 369–370. (doi:10.1016/j.tics.2009.06.007)
138 Saur, D. *et al.* 2008 Ventral and dorsal pathways for language. *Proc. Natl Acad. Sci. USA* **105**, 18 035–18 040. (doi:10.1073/pnas.0805234105)
139 Perani, D., Saccuman, M. C., Scifo, P., Anwander, A., Spada, D., Baldoli, C., Poloniato, A., Lohmmann, G. & Friederici, A. D. 2011 Neural language networks at birth. *Proc. Natl Acad. Sci. USA* **108**, 16 056–16 061. (doi:10.1073/pnas.1102991108)
140 Wilson, S. M., Dronkers, N. F., Ogar, J. M., Jang, J., Growdon, M. E., Agosta, F., Henry, M. L., Miller, B. L. & Gorno-Tempini, M. L. 2010 Neural correlates of syntactic processing in the nonfluent variant of primary progressive aphasia. *J. Neurosci.* **30**, 16 845–16 854. (doi:10.1523/JNEUROSCI.2547-10.2010)

141 Tyler, L. K. & Marslen-Wilson, W. 2008 Fronto-temporal brain systems supporting spoken language comprehension. *Phil. Trans. R. Soc B* **363**, 1037–1054. (doi:10.1098/rstb.2007.2158)
142 Tyler, L. K., Marslen-Wilson, W., Randell, B., Wright, P., Devereux, B. J., Zhuang, J., Papoutsi, M. & Stamatakis, E. A. 2011 Left inferior frontal cortex and syntax: function, structure and behaviour in patients with left hemisphere damage. *Brain* **134**, 415–431. (doi:10.1093/brain/awq369)
143 Rilling, J. K., Glasser, M. F., Preuss, T. M., Ma, X., Zhao, T., Hu, X. & Behrens, T. E. 2008 The evolution of the arcuate fasciculus revealed with comparative DTI. *Nat. Neurosci.* **11**, 426–428. (doi:10.1038/nn2072)
144 Brauer, J., Anwander, A. & Friederici, A. D. 2011 Neuroanatomical prerequisites for language functions in the maturing brain. *Cereb. Cortex* **21**, 459–466. (doi:10.1093/cercor/bhq108)
145 Teinonen, T., Fellman, V., Naatanen, R., Alku, P. & Huotilainen, M. 2009 Statistical language learning in neonates revealed by event-related brain potentials. *Bmc Neurosci* **10**, 21. (doi:10.1186/1471-2202-10-21)
146 Friederici, A. D., Mueller, J. L. & Oberecker, R. 2011 Precursors to natural grammar learning: preliminary evidence from 4-month-old infants. *PLoS ONE* **6**, e17920. (doi:10.1371/journal.pone.0017920)
147 Endress, A. D., Carden, S., Versace, E. & Hauser, M. D. 2010 The apes' edge: positional learning in chimpanzees and humans. *Anim. Cogn.* **13**, 483–495. (doi:10.1007/s10071-009-0299-8)
148 Endress, A. D., Nespor, M. & Mehler, J. 2009 Perceptual and memory constraints on language acquisition. *Trends Cogn. Sci.* **13**, 348–353. (doi:10.1016/j.tics.2009.05.005)
149 Joshi, A. 2003 Tree-adjoining grammars. In *Oxford handbook of computational linguisitcs* (ed. R. Mikkov), pp. 483–501. New York, NY: Oxford University Press.
150 Comon, H., Dauchet, M., Gilleron, R., Jacquemard, F., Lugiez, D., Löding, C., Tison, S. & Tommasi, M. 2007 Tree automata techniques and applications. See http://tata.gforge.inria.fr/.
151 Ferreira, F., Lau, E. F. & Bailey, K. G. D. 2004 Disfluencies, language comprehension, and Tree Adjoining Grammars. *Cogn. Sci.* **28**, 721–749. (doi:10.1207/s15516709cog2805_5)
152 Hagoort, P. 2005 On Broca, brain, and binding: a new framework. *Trends Cogn. Sci.* **9**, 416–423. (doi:10.1016/j.tics.2005.07.004)
153 Joshi, A. K. 2004 Starting with complex primitives pays off: complicate locally, simplify globally. *Cogn Sci.* **28**, 637–668. (doi:10.1207/s15516709cog2805_2)
154 Chater, N. & Manning, C. D. 2006 Probabilistic models of language processing and acquisition. *Trends Cogn. Sci.* **10**, 335–344. (doi:10.1016/j.tics.2006.05.006)
155 Johnson, M. & Riezler, S. 2002 Statistical models of syntax learning and use. *Cogn. Sci.* **26**, 239–253. (doi:10.1207/s15516709cog2603_2)
156 Klein, D. & Manning, C. D. 2004 Corpus-based induction of syntactic structure: models of dependency and constituency. In *ACL '04, Proc. 42nd Annual Meeting on Association for Computational Linguistics, Barcelona, Spain, 21–26 July 2004*, no. 478. Stroudsburg, PA: Association for Computational Linguistics.
157 Buhrmester, M., Kwang, T. & Gosling, S. D. 2011 Amazon's mechanical turk: a new source of inexpensive, yet high-quality, data? *Perspect. Psychol. Sci.* **6**, 3–5. (doi:10.1177/1745691610393980)
158 Petkov, C. I. & Wilson, B. 2012 On the pursuit of the brain network for proto-syntactic learning in non-human primates: conceptual issues and neurobiological hypotheses. *Phil. Trans. R. Soc. B* **367**, 2077–2088. (doi:10.1098/rstb.2012.0073)
159 Ballard, D. 1999 *An introduction to natural computation*. Cambridge, MA: MIT Press.
160 Richards, W. (ed.) 1988 *Natural computation*. Cambridge, MA: MIT Press.
161 Rolls, E. T. & Deco, G. 2001 *Computational neuroscience of vision*. Oxford, UK: Oxford University Press.
162 Siegelmann, H. T. 1995 Computation beyond the Turing limit. *Science* **268**, 545–548. (doi:10.1126/science.268.5210.545)