Knots, Language, and Computation: A Bizarre Love Triangle? Replies to Objections

Sergio Balari, Antonio Benítez-Burraco, Marta Camps,

Víctor M. Longa & Guillermo Lorenzo

Die Philosophie löst Knoten auf in unserm Denken; daher muß ihr Resultat einfach sein, das Philosophieren aber so kompliziert wie die Knoten, welche es auflöst.[1]
L. Wittgenstein, *Zettel*, § 452

1. Map of the Problematique (Some Background)

In a series of papers we have been developing a proposal for a novel methodology to 'read' the archaeological record in order to overcome a number of problems posed by the reliance of contemporary palaeoanthropology on such ill-defined notions as 'modern behavior' and 'symbolic culture' (Balari *et al*. 2008, 2011; see also Benítez-Burraco *et al*. 2008 and Balari *et al*. 2010 for some additional background). Our proposal is based on the idea that a formal analysis of the material remains left by our ancestors may prove useful in determining the kinds and amount of cognitive resources deployed to produce such objects, in other words, that manufactured objects are transparent with respect to the biological structures underlying the processes necessary to produce them. By performing such an analysis, we contend, one is capable of inferring the computational complexity of the said cognitive tasks and to advance hypotheses concerning the

[1] "Philosophy unties knots in our thinking; hence its results must be simple, but philosophizing has to be as complicated as the knots it unties" (translation by G.E.M. Anscombe).

architecture of the mind capable of performing them. Our hypotheses were framed against the background of a general model of the architecture of minds we only sketched in the papers referred to above, but which cross-fertilized with a parallel proposal to apply the same methodology in the context of the study of animal cognition in general (Balari & Lorenzo 2008, 2009, to appear).

This more general framework relies on the contention that animal minds / brains are all constructed following a very similar pattern. This claim is mostly grounded in the fact that developmental systems tend to follow rather conservative pathways and easily fall into canalization patterns (Waddington 1957) or follow certain developmental inertias (Minelli 2011 and Striedter 2005 for comparative developmental data on vertebrate brains), but also finds support on a number of neuroanatomical, molecular and other kinds of data we'll try to spell out in the paragraphs to follow.

The main architectural component in our model is a core engine or natural computational system (NCS) subserving some (but not necessarily all) of the main cognitive functions of an animal mind, including low-level ones like motor planning and execution. This NCS is a serial, digital, von Neumann machine implemented on an analog, parallel and continuous substrate,[2] which may be modeled by an abstract machine or automaton in the sense of the mathematical theory of formal languages and automata. We call these abstract models 'computational phenotypes' in order to make explicit the idea that computational activity is a phenotypic trait that one can associate to certain specific neuro-anatomical configurations. Computational phenotypes are thus abstract characterizations of the basic models of computation implemented by the said neuro-anatomical configurations, and since phenotypic differences are eminently structural in nature, it makes sense to capture the differences between computational phenotypes by appealing to differences in structure between models of computation. Therefore, the natural place to look at in order to provide an abstract structural characterization of a model of computation is automata theory as developed in connection with formal language theory. We will have more to say about formal language theory in section 2, especially with respect to its relation to the theory of computational complexity, but for the time being suffice it to say that it is concerned with the structural complexity of 'languages', where 'language' here is a technical term referring to sets specified as collections of strings made up of symbols taken from a finite alphabet. Thus, for example, the set of integers $\mathbb{Z} = \{\dots, -1, 0, 1, \dots\}$ may be represented in binary by the language made up of strings over the alphabet $\Sigma = \{0, 1\}$ as $\mathbb{Z} = \{\dots, 10000001, 00000000, 00000001, \dots\}$, for an 8 bit system.[3]

---

[2]   The idea that the brain is an analog, parallel processor that nonetheless implements serial, digital processes is supported by evidence coming from different fields, such as neural computation (Sarpeshkar 1998, 2009), neurobiology (Alle & Geiger 2006, Shu *et al.* 2006) and neuropsychological models (Zylberberg *et al.* 2011). This, in fact, adds an extra dimension of variation to the ones to be proposed below, namely the possibility that the neural computations underlying certain behaviors follow the analog, parallel path instead of the digital, serial one. We will have nothing to say about that in this paper, but see Balari & Lorenzo (to appear: chap. 8) for discussion.

[3]   Unless otherwise stated, and in addition to Chomsky's original papers cited in the text, our basic references in the following informal presentation of language and automata theory are

Since its original development by Noam Chomsky in the late 1950s and early 1960s (Chomsky 1959; 1963), the theory's main goal has been that of classifying languages (in the technical sense just defined) in a hierarchy of increasing complexity, the so-called Chomsky Hierarchy. Traditionally, the Chomsky Hierarchy is assumed to be organized into four classes (sets of languages) related by proper containment as follows: Type3 ⊂ Type2 ⊂ Type1 ⊂ Type0. Type3 (regular) languages are the structurally simplest ones and Type0 (recursively enumerable) languages the more complex ones, with Type2 (context-free) languages and Type1 (context-sensitive) languages sitting in between these two extremes of the complexity spectrum.

Now, given that languages are just (possibly infinite) sets of strings defined over a finite alphabet, they incorporate no direct record of their complexity, but, as Chomsky showed, this can be assessed indirectly by studying the complexity of the finite devices capable of specifying them — grammars and automata. As for grammars, different degrees of complexity are obtained by successively imposing constraints on the format of rules to go from unrestricted grammars for recursively enumerable sets to right- (or left-) linear grammars for regular sets. In the case of automata, the difference in structure follows from the constraints on the working memory space the device has at its disposal to perform the computation, with Turing machines having infinite space (and time) resources to work with and finite-state automata having no working memory space at all. This focus on memory space may seem unjustified at first blush, because, unlike the case of grammars, the traditional characterization of automata is not easily seen as a series in which each automaton is defined as an extension of the immediately preceding one. Thus, whereas the pushdown automaton is just like a finite-state automaton with a memory stack (plus the minimal adjustments to its finite control unit to be able to manipulate the stack), the linear-bounded automaton for context-sensitive languages is a Turing machine whose memory working space is constrained by the length of the input string, that is, it can only use the space in the input tape already occupied by the input string. This is clearly a constraint on memory, but it is hard to see how it relates to the structural properties of the stack of a pushdown automaton. This relation only became obvious after the work of K. Vijay-Shanker and David Weir (Vijay-Shanker & Weir 1994, Weir 1992, 1994) who, building on original work by Nabil Khabbaz (1974), showed that a language hierarchy can be defined with context-free languages being the first in the hierarchy and extending into the space of what traditionally fell under the label of Type 1 languages.[4]

Vijay-Shanker and Weir's results came together with the demonstration that the pushdown automaton, the corresponding machine model for context-free grammars, is just a particular instance of a more general class of automata using pushdown storage and differing only in the structure of the storage mechanism. Thus, on the basis of these results, we could conceptualize the universe of languages as a space organized into three properly contained sub-hierarchies of

---

Hopcroft & Ullmann (1979) and Lewis & Papadimitriou (1981).

[4]   As far as we can tell, it remains open the question whether this sub-hierarchy includes all recursive languages in addition to just the context-sensitive ones and excludes the recursively enumerable languages that are not recursive.

increasing structural complexity:

(i)   The Regular Hierarchy
(ii)   The Pushdown Hierarchy
(iii) Unrestricted Systems

Thus, the Pushdown Hierarchy contains all those languages that, like the context-free languages, can only be recognized by an automaton making use of some kind of pushdown storage. Similarly, the Regular Hierarchy contains the so-called Subregular Hierarchy (Rogers & Hauser 2010, Rogers & Pullum 2011) and includes, at its maximal level of complexity, the regular or finite-state systems. The general machine model for the Regular Hierarchy is the finite-state automaton. These systems are devices capable of performing computations — often very complex ones — without resorting to any external storing device. Finite-state automata are, therefore, systems with no memory whose computational power relies exclusively on the (finite) number of states the machine can be in. Pushdown systems, on the other hand, do possess an external storage device (the so-called pushdown stack), which they can manipulate to help and drive the steps of a computation. Complexity within the Pushdown Hierarchy, where Type 2 and Type 1 systems are properly included, increases as the structure of the pushdown stack is made richer and stack embeddings are allowed within stacks, making it possible to perform more and more complex computations without actually altering the basic processing regime of the whole device. Let us emphasize the fact that the only difference between a finite-state automaton and a pushdown automaton is the presence of the memory stack in the latter, but, for the rest, both are essentially the same kind of device, with the minimal adjustments in the finite control unit of the pushdown automaton to allow for the operations of writing and erasing symbols from the stack (or stacks if it is of a more complex type).

Seen in this way, the complexity hierarchy more clearly illustrates that computational power is just a function of memory or, in other words, of the space resources a computational device has at its disposal. This is, according to our proposal, a first dimension of variation for NCSs. More concretely, we would like to suggest that NCSs are constructed on the basis of a very conservative core engine, always following the same processing regime, which can nevertheless be complemented by a working memory device. The presence/absence and sophistication of this working memory component will determine the computational capabilities of the system, such that a NCS with no memory will be less powerful than one with memory, and, within the class of NCSs with memory, those with complex and sophisticated memory systems will be more powerful than those with a more basic working memory.[5]

---

[5]   Leaving aside the case of unrestricted systems (Turing machines), which constitute a special case, our bipartite division between a regular and a pushdown hierarchy can also be motivated on computational complexity theory grounds. Thus, regular systems correspond to those languages in the set SPACE(k), where k is an integer, that is to those languages that can be decided in constant space or, in other words, to those languages for which the space resources used by the decision procedure are independent of the length of the input, which

Before we go on, it is important to point out that our use of the term 'working memory' has little to do with any psychological model of memory, as for example the Working Memory of Baddeley (2007) and to which Frederick Coolidge and Thomas Wynn (e.g., Coolidge & Wynn 2004, Wynn & Coolidge 2011) have referred in their work on the Neanderthal mind. Our computational phenotypes are, crucially, not performance models, but abstract characterizations of the models of computation over which a full-fledged performance model may be constructed. According to this, a vertebrate brain can thus be characterized in terms of its computational phenotype (Balari & Lorenzo 2008, 2009, to appear), a characterization in complexity terms focusing on the computational model of the NCS that the said brain implements. Thus, if the analysis of some form of behavior reveals that the complexity of the computations necessary to perform the task is equivalent to a finite-state automaton, we can conjecture that the computational phenotype of the NCS implemented in the brain of this creature is, at least, of the regular type; similarly with the other possible computational phenotypes, with the one corresponding to the human NCS being capable of processing language and hence sitting, at least, within the mildly context-sensitive region of the complexity space.

In some sense, then, our proposal can be seen as an extension of an experimental paradigm originally initiated by W. Tecumseh Fitch and Marc Hauser, based on the aural pattern-recognition abilities of cotton-top tamarins and reported in Fitch & Hauser (2004); see also O'Donnell *et al.* (2005) and Rogers & Hauser (2010). The same paradigm was later applied to starlings, with seemingly equal success by Gentner *et al.* (2006), although it has been subjected to several criticisms from different quarters (e.g., Perruchet & Rey 2005, Heijningen *et al.* 2009, Petersson *et al.* 2010, Rogers & Pullum 2011). The main difference between what we are proposing here and the assumptions of aural pattern-recognition experiments is that instead of focusing on learning capabilities our suggested methodology proceeds in the reverse direction, from the behavioral patterns the subject is naturally capable to perform to their structural complexity and, from there, to the minimal model of computation required for the task, and it is thus immune to the shortcomings observed by, for example, Rogers & Pullum (2011) that affect aural pattern-learning experiments.

Thus, what we have characterized here as a NCS roughly corresponds to what Hauser *et al.* (2002) called the Faculty of Language in the narrow sense (FLN), but our characterization is broader in the sense that we do not conceive of our NCS as either language-specific or human-specific. Indeed, even though we follow Hauser *et al.*'s (2002) methodological proposal of distinguishing between narrow and broad aspects of some cognitive ability, we deny to their FLN its human and language specificity, whence our terminological switch here to NCS.[6]

---

is tantamount to saying that no memory is required in the process. For all languages in the pushdown hierarchy, on the other hand, the space resources spent during decision will be some function of the length of the input, and consequently some storage mechanism is required. We come back to this in section 2, but see Papadimitriou (1994) for details.

[6]   Something that — we are ready to acknowledge this — we should've done before in order to avoid misunderstandings stemming from our infelicitous use of FL in contexts in which the term was clearly inappropriate.

Now, from all this it follows that NCSs are functionally unspecific devices, i.e. not specially tailored to perform one or another cognitive task. NCSs just are capable of executing a recursive procedure capable of constructing more or less complex representations as a function of the input and the memory resources available.[7] NCSs therefore, in order to be operative, need to be connected to some external modules supplying their input and capable of receiving their output. Seen from this perspective, external modules impose a number of constraints on the workings of a NCS, mostly concerning certain properties having to do with the nature of the input and certain properties concerning the nature of the output, but, crucially, these constraints will have little, if anything, to say about certain structural properties of the objects produced by the NCS, since these will critically depend on the memory resources available to the recursive procedure implemented by the NCS. There is thus a tension between the interface conditions imposed by the external systems and the computational capabilities inherent to the NCS which defines a two-dimensional space of variation, with one of the axes corresponding to the working memory space the NCS has access to and the other axis corresponding to the number and kind of the external systems the NCS interfaces to. Thus, following Hauser *et al.* (2002), the minimal architectural requirements for human language are a NCS with a computational power at least equivalent to that of a mildly context-sensitive system interfaced to a Sensory-Motor module and a Conceptual-Intentional module.

Putting now this whole view in a broader perspective, note that this model opens up new avenues of research within the field of cognitive archaeology in particular and cognitive science in general, since, given the two dimensions of variation we envisage here, a number of different possible configurations for 'other minds' can be imagined. For example, it is possible that a linguistic phenotype has existed with the same external systems as the human one but interfaced to a less powerful NCS; or one lacking some of the interfaces necessary for the full externalization of thought; or one simply not yet fully satisfying the interface constraints imposed by the external systems; similarly for other cognitive abilities different from language, and as already suggested above, for other cognitive abilities in other animal species. The question, of course, is eventually an empirical one, and the methodology we are proposing here can be, in our opinion, a useful tool for clarifying all these questions.

The general model we've just outlined in the foregoing discussion is not merely a product of our speculations, but, as we pointed out at the beginning of this section, there exist numerous pieces of evidence coming from several other

---

[7]   The issue of 'recursion' has become a debated question since, at least, the publication of Hauser *et al.* (2002). Here, following Tomalin (2007, 2011), we will adhere to the terminological distinction between 'recursion' and 'self-similar embedding'. The first term refers to the property, common to a number of computational devices, of being able to take as input a number of objects of some specific type in order to produce a new object or set of objects of the same type which in turn may later feed a subsequent step in the computation to produce new objects of the same type and so on. The second one refers to a structural property observed in the objects produced by certain computational devices with the appropriate memory resources (minimally, a push-down stack), like, for example, expressions in a natural language. Thus, to get objects with self-similar embedding some recursive procedure must be applied, but not all applications of a recursive procedure necessarily yield objects with the property of self-similar embedding.

areas of research that seem to point in the same direction. What follows is a brief delineation of this evidence.

To begin with, there exists abundant clinical evidence suggesting the comorbidity among diverse disorders entailing deficits of diverse sort that can be yet construed as equivalent in computational terms. For instance, people affected by Williams-Beuren syndrome exhibit linguistic (Karmiloff-Smith 2006) and visuo-spatial deficits (Hudson & Farran 2010), which are interpretable in terms of processing problems with context-free and context-sensitive operations. Similarly, the prevalence of drawing deficits among dyslexics is a well-known fact, which invites to explain their state in terms of a visuo-constructive deficit (Eden *et al.* 2003, Lipowska *et al.* 2011), or even better, of a more general deficit affecting the rule abstraction mechanism inherent to sequential learning, which would impair both linguistic and visuo-motor abilities (Vicari *et al.* 2005, Pavlidou *et al.* 2010). Finally, it is also commonly observed that language and motor deficits co-occur developmentally (Webster *et al.* 2005, Cheng *et al.* 2009, Iverson & Braddock 2011; see Rechetnikov & Maitra 2009 for a meta-analysis), implying that voluntary motor actions entail diverse motor primitives or 'movemes' (Del Vecchio *et al.* 2003) arranged in different ways according to specific combinatorial or syntactic rules (see Flash & Hochner 2005, and references cited therein). All this well-attested comorbidity is easily explained if one and the same computational device is affected.

Current neurobiological research provides us with the most compelling evidence for the plausibility of our model. For example, dissimilar linguistic modalities, such as sign and spoken languages, otherwise equally complex in structural terms (Brentari *et al.* 2010), are processed by similar brain mechanisms, as attested by numerous neuro-imaging and lesion studies (MacSweeney *et al.* 2008, among many others). This evidence gives support to the idea that the NCS is unspecific in functional terms, but capable of coupling to different modules for exteriorizing its outputs. Besides, the brain seems to remain on basic neural 'binding mechanisms' (like cortical synfire chains) to generate any kind of composite objects at the representational level (Flash & Hochner 2005: 663), and there is ample neuro-imaging evidence supporting the hypothesis of different kinds of computations being performed by a 'central' device (see Dipietro *et al.* 2009 with regards to drawing). Eventually, fMRI studies show that similar patterns of activation arise in response to diverse tasks if they are computationally equivalent, ultimately suggesting that the same brain areas (and plausibly, the same computational device) are involved. For instance, drawing bilaterally activates a wide network of subcortical and cortical structures (Makuuchi *et al.* 2003), many of which are also involved in language processing (see Makuuchi 2010 for a review).

Although the precise topography of the neural substrate of such computational device is somehow controversial, it plausibly relies upon the coordinated activities of diverse brain areas, both cortical and subcortical. For instance, according to Lieberman (2000, 2002, 2006) language is tantamount to a computational device capable of processing symbolic elements, with this device conceivably being the outcome of the interaction between a sequencer (the activity performed by the basal ganglia) and a working memory (the activity

executed by diverse cortical structures). In a similar fashion, Ullman (2001, 2004) has hypothesized that a procedural memory exists (ultimately emerging from the coordinated activities of a complex network of cortical, subcortical, and cerebellar neuronal populations) that allows for the existence and functioning of "the mental grammar, which subserves the rule-governed combination of lexical items into complex representations" (Ullman 2004: 231). Crucially, both Lieberman's computational device and Ullman's procedural memory are ultimately conceived as to subserve the learning and execution of diverse tasks, both linguistic and non-linguistic, so when this neural architecture is damaged an admixture of both linguistic and non-linguistic symptoms are observed in affected people.

This is the case, for instance, of Huntington disease, a neurodegenerative condition in which a defective variant of the HD protein accumulates in the cell nucleus and cytoplasm, specifically killing the gabergic neurons of the caudate (Gusella & MacDonald 2006), and thus plausibly disturbing the neural substrate of the sequencer in our model. In affected people pervasive problems with the application of rules are observed during language processing; however, the disease also encompasses a defective processing ability of motor routines, as well as other diverse cognitive deficits and psychiatric disturbances (Gusella & MacDonald 2006). Other neurological pathologies affecting the basal ganglia exhibit quite similar phenotypic profiles. For example, Parkinson disease is also caused by a selective damage of some components of this subcortical region, to be precise, the dopaminergic neurons that project to the *substantia nigra* (Surmeier *et al.* 2010). Once again, linguistic and motor deficits regularly co-occur in affected people (Grossman *et al.* 1991, Duffy 2005). In the whole, we recurrently observe that language disabilities correlate to a variety of neuropsychological and motor changes in all these conditions (see Murray 2000, for a review of Huntington and Parkinson diseases).

The abnormal development of this neural circuitry through mutations of any of the genes involved in its organization also renders symptoms that are not domain-specific. A classic (but still illustrative) instance is the first mutation to be identified on *FOXP2*. In people bearing the R553H substitution, the primary pathology is located in the caudate, but other brain regions, otherwise relevant for language processing, are also structurally and/or functionally impaired (Vargha-Khadem *et al.* 1998, Watkins *et al.* 2002a, Belton *et al.* 2003, Liégeois *et al.* 2003). In fact, there is ample evidence supporting some key role of *FOXP2* in modulating the development and functioning of (specific) cortico-thalamic-striatal circuits associated to motor planning, sequential tasks, and procedural learning (for a review, see Marcus & Fisher 2003, Vargha-Khadem *et al.* 2005, Fisher & Scharff 2009), a brain network which could plausibly match the neural substrate of our NCS. Although there is a hot dispute around the precise phenotypic profile (and the underlying deficit) of the disorder linked to the mutation of the gene, motor and linguistic deficits are simultaneously observed in affected people (Gopnik 1990, Vargha-Khadem *et al.* 1995, Watkins *et al.* 2002b, Vargha-Khadem *et al.* 2005, Shriberg *et al.* 2006), thus precluding this condition from being merely characterized as a speech or even a (specific) language disorder. Animal models reinforce the functional unspecificity of the neural circuitry *FOXP2* contributes to (Shu *et al.* 2005, Fujita *et al.* 2007).

Finally, acquired damages of important constituents of the neural substrate of our NCS also give rise to deficits of diverse condition, eventually reinforcing the functional non-specificity of the device. For example, a focal damage of the left basal ganglia (in particular, of certain frontal-subcortical circuits) symptomatically manifests as a decreased ability in both verbal and visual modalities (Troyer *et al.* 2004), and both limb movements and language are impaired when cortico-basal ganglia-thalamo-cortical circuits are damaged, as observed in drawing, overwriting and repeating words and phrases in absence of external commands (Fung *et al.* 1997).

This is just a sample we've recently compiled, which could easily be extended with further data, but one that clearly suggests that evidence from neurobiology is amenable to an interpretation entirely compatible with the model sketched here.

That said, we let's turn now to the objections.

2.   The Correct Use of Soap (and a Conclusion)

Now that we have provided a detailed account of our ideas, we are in a better position to address David J. Lobina's criticisms. We will start, however, with some general comments on a point Lobina does not directly touch on in his note, but which remains implicit in most of what he says, so we deem it necessary to get into that before turning to other issues.[8]

A key feature of our proposal is that the processing capabilities connected to FL in humans are independent from any domain of conceptual primitives, even those dedicated to belief fixation, planning, and the like ('thought', roughly speaking) that, according to recent minimalist theorizing, conflate with FL into a unitary domain (Hinzen 2006, Chomsky 2007). Notwithstanding the fact that Lobina refrains from discussing general features of mental architecture in his paper, a clarification of our particular view of the computational mind is in order, as he has elsewhere referred to it as "bizarre", "incoherent", and "strange" (Lobina 2012) and continues to cast doubt on its soundness in the *Biolinguistics* piece (this issue, p. 76), attributing to us assumptions that we do not really share.

As a matter of fact, we do share certain background assumptions with Lobina, to wit: (1) That thoughts are *bona fide* (i.e. contentful) representations; and (2) that contentful representations are language-like entities — i.e. something along the lines of Fodor's 1975 LOT hypothesis (Lobina 2012: 2–3).[9] We think that these are reasonable assumptions, but not because their truth is somehow warranted: Thoughts could happen to be some kind of brute-causal associationist networking, instead of contentful representations, and contentful representations could happen to be map-like, instead of language-like, entities. Yet, as observed for example by Devitt (2006), (1) and (2) are the most successful hypotheses thus far in predicting other people's behavior as well as in explaining the productivity and systematicity of thought, among other things; in other words, it is not their

---

[8]   All references to Lobina's work including only a page number are to his paper "All tied in knots", published in this issue of *Biolinguistics*.

[9]   This other paper by Lobina is not paginated, so page numbers in our references to it should be interpreted as 'first page', 'second page', and so on.

purported truth what makes them respectable, but rather their explanatory force. We share this set of convictions with Devitt, as we think Lobina does.

So we accept (with Lobina) the idea of an autonomous thought domain, whose primitives are easily accessible for computations that map them into complex thoughts. However, the thesis that the processing competence that we use in composing meaningful expressions (say, sentences) is inextricably linked to the existence of rich contentful, language-like thoughts (which Lobina mistakenly attributes to us) is not in our opinion a favored hypothesis on plausibility grounds similar to that of (1) and (2). Thus, contrary to the adoption of the representational theory of mind and the LOT hypothesis as default premises given the state of the art of cognitive science and philosophy of mind, a similar move is far from warranted with respect to the question of intrinsically connecting the processing capabilities of mind with the kind of concepts and intentions that it is able to deal with. The clearest statement in this direction that we are aware of presents this processing competence as an adaptation to the evolved function of composing (and subsidiarily expressing) complex thoughts (Devitt 2006, Chomsky 2010). We think, however, that adaptationism provides no sufficient grounds for favoring such a hypothesis, for reasons that are explained at length in Fodor & Piattelli-Palmarini (2010). Here we just want to remember that George C. Williams, one of the most brilliant evolutionary biologists of the 20th century and an adaptationist himself, defended that adaptations, far from being seen as default explanations, should be treated as explanatory last resorts. The reason is simple: The long, cumulative and highly complex chains of events leading to true adaptations are not to be expected to occur very frequently (Williams 1966). So the idea that a processing competence exists as an adaptation for the elaboration and expression of thoughts is not one to be contemplated in the absence of very (very) strong evidence and, as a first move, we should reject it.

The impact of these observations on establishing and exploring a reasonable view of the architecture of mind is in our opinion clear: The thesis that there exists a processing competence inextricably and exclusively connected to the rich conceptual competence in charge of belief fixation and the like looses all the beforehand motivation that it has under the umbrella of adaptationism; and so, we become free to explore the alternative idea that the said competence has an independent evolutionary history, not linked to any conceptual, sensory or motor domain in particular, as well as the complementary idea that it has gained access to different cognitive domains throughout hominid evolution (or evolution without more qualifications, as it could happen to be a very ancient trait; see Balari & Lorenzo 2008, 2009, to appear). The idea boils down to the supposition that humans make complex beliefs for the same reasons that they make complex arguments, complex paintings, complex poems or complex knots (which is not to say that knots, poems, paintings, and especially arguments are always complex). In other words, we support the idea of decoupling linguistic competence into autonomous components, in the spirit of Hauser *et al.* (2002) save the important detail that we envision its core processing component (i.e. the NCS) as not language and not even human specific. We honestly wonder what is so weird about this idea. Its truth is not warranted, of course (as almost everyone else's

ideas in the field), but it opens paths worth being explored (and more clearly amenable to empirical testing than most theses in the field). Are we not Fodorian enough? Not neo-Chomskyan enough (perhaps to Lobina's surprise)? Granted, but to this we can only reply one thing: There's life beyond authorities.

Therefore, it is against this general background that must be construed our claim that some important aspects of knot-tying abilities might be parasitic on language in humans. According to our view, whatever cognitive modules participate in knot-tying abilities crucially share the same computational basis with those participating in linguistic abilities. We will immediately turn to knots, but, before, we'll need to go back to automata, performance models and computational complexity.

Lobina's criticisms are articulated along two main themes: Our purported misuse of complexity results deriving from formal language theory and the theory of computational complexity, on the one hand, and what in his opinion is an illegitimate appeal to topology in our claims concerning knotting abilities in humans, on the other. Both points are eventually connected, but we shall respond to them in turn, and only towards the end will we be able to tie all the threads together.

As for the first point, Lobina pretends that we are mixing questions of weak generative capacity with questions of strong generative capacity and leaping from there to illicitly inferring that natural language has a specific inherent computational complexity. It appears then that some steps in the argument have escaped Lobina's attention, so we'll try to spell them out in full.

Recall that the Chomsky Hierarchy classifies languages according to their structural complexity and that such complexity can only be assessed through the devices that are capable of specifying one or another language. Thus, if the structural complexity of a set can only be assessed with respect to one or another model of computation, it should not come as a surprise that Chomsky has always insisted on the fact that what is important in the study of a language (*any* language, natural languages included) is its grammar, not just the strings that make it up. This observation is at the core of the whole generative linguistics enterprise, since, as also pointed out by Chomsky, there are many possible different grammars one can think of capable of generating exactly the same stringset; we shall say in this case that these grammars are all 'weakly equivalent' or that they have the same 'weak generative capacity'.

The challenge, when our focus of interest is natural language, is, therefore, which of all the imaginable weakly equivalent grammars is the one that really captures the actual structure of natural language expressions or, using Chomsky's own words, which grammar is the descriptively adequate one. Note that descriptive adequacy is defined not just with respect to weak generative capacity but, rather, with respect to a stronger condition incorporating the notion of structural description. Thus, two weakly equivalent grammars are not necessarily also 'strongly equivalent', i.e. they do not necessarily assign the same structural descriptions to the strings of a set or have the same 'strong generative capacity'. Note that the notion of strong generation somehow transcends the notion of model of computation, since grammars and automata as defined in formal language theory, are only weak generators, not strong ones, they do not

assign structural descriptions to strings in a set.

Importantly, though, this does not make all the complexity results presented in the preceding discussion irrelevant, because another crucial outcome of Chomsky's original work is that, whatever the descriptively adequate grammar for natural language eventually turns out to be, its power will be beyond that of context-free grammars, and, as shown in later work also reported above, it most probably lies within the power of a mildly context-sensitive one. These are "the limits of finite-state description" (and of context-free description) that according to Chomsky (1956, 1957) force linguistic theories to turn to more powerful devices than finite-state and context-free grammars. Thus, the only reason for saying that the grammar of a natural language is some mildly context-sensitive grammar is precisely that only a grammatical formalism of this power will be able to generate the appropriate structural descriptions. Note, however, that these complexity results refer to the inherent structural complexity of natural language and are therefore independent from the kind of grammatical formalism we may want to favor as our theory of linguistic competence. They set a lower bound of complexity in the sense that an adequate grammatical formalism should possess at least the same expressive power of a mildly context-sensitive grammar. Now, since for each grammar there is a corresponding automaton, it doesn't make much difference on which perspective we want to put the emphasis when concerned with such inherent structural complexity of language.

Moreover, as the preceding discussion suggests, we must assume that a type-token relation exists between a grammar of a specific type (say, a context-free grammar) and a grammar generating some specific language (say, some context-free language); similarly with automata. Our computational phenotypes are types precisely in this sense, and, consequently, to repeat, they are not — cannot be — performance models, just basic specifications from which performance models can be built. Thus, if we ever wanted to build a parser P (a strong generator) for some language given some grammar G, we should base our construction on the appropriate model of computation A (a weak generator), because parsing presupposes recognition. That much seems to have escaped Lobina's attention, as he constantly insists on stating that automata are just recognizers and not generators, but as Chomsky (1963: 332) already made clear a long time ago:

> "It is immaterial whether we picture an automaton as a source generating a sentence symbol by symbol as it proceeds from state to state or as a reader switching from state to state as it receives each successive symbol of the sentence it accepts."

And, if any doubt was left about the fact that parsing presupposes recognition, take the following quote from Earley's (1970: 95) original paper describing his popular parsing algorithm for context-free grammars:

> "A *recognizer* is an algorithm which takes as input a string and either *accepts* or *rejects* it depending on whether or not the string is a sentence of the grammar. A *parser* is a recognizer which also outputs the set of legal derivation trees for the string." (Emphasis in the original.)

Thus, since our proposal is clearly noncommittal to any specific model of grammar, a large part of Lobina's subsequent observations concerning grammar formalisms and parsers constitute a blatant *non sequitur* that only adds confusion and deviates the discussion form the real point. And the point is, in this case, that anyone interested in building a performance model should pay attention to these structural facts concerning the complexity of human language in order to be successful. But not just to these, because there are also questions of computational complexity that need to be taken into account. Enter the theory of computational complexity, then.

Before getting into the details, it is crucial to make clear that both the Chomsky Hierarchy and the theory of computational complexity are concerned with languages and with their classification into classes.[10] Of course, the criteria for their classification are different in each case, but it is important to see that some language that falls in some class in the Chomsky Hierarchy will also show up in some computational complexity class and therefore connections can be established between the inherent structural complexity of this language and its inherent computational complexity. Thus, as we already pointed out in footnote 5 above, the languages that fall within the regular class in the Chomsky Hierarchy are exactly the ones conforming the SPACE(k) class of computational complexity theory, therefore we know that any two regular languages (i.e. of identical structural complexity) have also the same computational complexity. This point is critical, because anyone failing to appreciate this might fall in the trap of believing that we are juggling with complexity classes and jumping from one perspective to the other and back without much justification. We're afraid we were jumping too fast, so let's proceed at a slower pace.

Computational complexity theory is concerned with sets but it looks at them as if they were problems. As with any problem, we want to know if there's a solution forthcoming in the near future and, in this case, how hard it is to solve it.[11] The point here is that any problem we may think of can be represented as a language and, accordingly, that everything reduces to analyzing the complexity of language recognition problems. Since structural complexity is here not an issue, there's no need to consider different models of computation, indeed, the strategy is to fix a specific model of computation and to see how it behaves when dealing with different problems. The model of computation of choice for computational complexity theory is the k-string Turing machine, which can however be set to operate at different modes of computation — deterministic and nondeterministic — in order to establish certain complexity measures.

Finally, complexity is defined as the amount of time and/or space resources spent by the machine in order to solve the problem, where time is defined as the number of steps needed to solve the problem and space as the

---

[10]   Our basic references for this presentation of computational complexity theory are Lewis & Papadimitriou (1981) and Papadimitriou (1994).

[11]   Therefore, the universe of sets the theory of computational complexity deals with very much transcends that of the Chomsky Hierarchy, since it sometimes discovers that some sets represent utterly intractable problems, that are not even recursively enumerable. Remember that the Chomsky Hierarchy includes only all the recursively enumerable sets, but not the non-recursively enumerable ones.

number of cells in the tape visited by the machine during the computation. There's a twist here, however, that needs to be made explicit to make complexity results more understandable. Turing machines are very powerful devices and, in principle, they have unlimited amounts of time and space to perform a computation and, since when concerned with problems we also want to know if they can be solved efficiently, we need to put some constraints to this unlimited capacity of Turing machines. For this reason, computational complexity theory is actually concerned with defining the time and/or space bounds within which a problem can be solved efficiently. The usual practice is to determine the upper bound, i.e. the worst case, beyond which efficiency severely degrades, but it is also possible to determine a lower bound of complexity for a problem, such that if we are able to determine that the lower bound for a problem is in complexity class C, we know that the problem is at least as hard as any of the hardest problems in C (it could be harder), and we say that the problem is C-hard.

Now, back to complexity measures in terms of time and space, these are defined as functions on the length of the input string telling us the rate at which time or space grow as the length of the input grows, expressed $f(n) = O(g(n))$, where n is the length of the input. There are different possible functions of this kind, but the ones that most concern us here are $O(n^c)$ and $O(c^n)$, where c is a constant in both cases, and referred to as polynomial and exponential, respectively. Thus, when we say that an algorithm runs in time $O(n^3)$, as it is the case with Earley's (1970) parser, for example, we are actually stating that, in the worst case, it will spend an amount of time equal to the cube of the length of the input, and we will classify it in the class TIME(n^3). Note that, in principle, there are infinitely many TIME classes like the preceding one since the exponent of the function can be any integer,[12] but all sharing the property of defining polynomial time bounds. The union of all these classes is the complexity class **P**. Given this definition, it is clear that **P** is in fact a hierarchy of polynomial time classes, with some classes including harder problems than others. The hardest problems in **P** are the **P**-complete problems, where a C-complete problem, C a complexity class, is a problem that is at least as hard as any of the hardest problems in C (i.e. it is C-hard) and it is known to be in C.[13] Moreover, **P** is a deterministic class, meaning that any problem within this class can be solved efficiently in poly-nomial time by a k-string Turing machine working in deterministic mode.

Nondeterministic time classes are a bit different. This is mostly due to the fact that non-determinism is still a poorly understood notion and that the definition of recognition is weaker for nondeterministic Turing machines than it is for deterministic ones (Papadimitriou 1994: chap. 2, for details). Given the fact that a nondeterministic Turing machine, at any point of the computation, has at least two choices to follow, time measures do not refer to all the possible steps in a single computation (these would be too many) and are calculated differently

---

12   Another assumption within computational complexity theory is that the exponents in polynomial functions are somehow 'well-behaved' in the sense that they never become too large to make the problem effectively intractable, even if polynomial.

13   Complete problems are thus 'model' problems since both their lower and their upper complexity bounds are defined by the class they belong in, e.g., a **P**-complete problem is at least as hard as any problem in **P**, but not harder.

assuming that at least one path yields to acceptance. The most important non-deterministic time class is **NP** (for Nondeterministic Polynomial) and, like **P**, is defined as the union of all nondeterministic classes with polynomial characteristic functions. The discovery that some problem is in **NP** means that the Turing machine will reach a solution following some path and that the correctness of the solution can be verified through a succinct certificate (or polynomial witness) in polynomial time. The succinct certificate is an external device that can be consulted every time a 'yes' state has been reached and that provides an efficient procedure to check the result.[14] As we will see presently, the existence of succinct certificates for all problems in **NP** is a datum that in some cases has been interpreted as having implications for cognitive science.

Turning briefly to space complexity classes, these are constructed in exactly the same way as time complexity classes, with the proviso that space is taken to be a more costly resource than time because it can be reused. In the case of space, then, polynomial functions identify very hard problems, close to intractability. For this reason, when dealing with space, logarithmic or linear bounds are preferred over polynomial bounds, although the class that will be of interest for us here is, precisely, **PSPACE** (for Polynomial Space), which is a deterministic class.

Finally, the three classes considered here are related by inclusion, composing the following hierarchy of increasing complexity: $P \subseteq NP \subseteq PSPACE$. Note that the inclusions are not known to be proper, meaning that the classes might turn out to be equal. Indeed, the question whether $P = NP$ is one of the most important unsolved problems in complexity theory.

Now, back to natural language, we've seen that, structurally, it is more complex than context-free but less than context-sensitive, with a structural complexity equivalent to that of a mildly-context sensitive language. Recall that these complexity results refer to sets and that these very same sets will appear as members of some computational complexity class represented as recognition problems. The problem CONTEXT-FREE RECOGNITION[15] is in **P**; the problem CONTEXT-SENSITIVE RECOGNITION is in **PSPACE**;[16] then, the problem MILD-CONTEXT-SENSITIVE RECOGNITION should fall somewhere in between, or within any of the two bounding classes. Note that this inference is entirely independent from any consideration concerning parsing, choice of grammatical formalism, or any other architectural or formal consideration about performance models. It is, if anything may be characterized in this way, a fact following from the inherent structural properties of the languages considered and from their analysis as recognition problems. Now, thanks to the work of Eric Sven Ristad, we can add some very interesting results to the previous inference that square perfectly with it.

Very briefly, but see Ristad's (1993) monograph for the details, Ristad

---

[14] Therefore, the definition of **NP** that Lobina gives his note is, to say the least, non-standard, when he writes that "the algorithm will define multiple ways of processing the input without specifying which one it will take, in polynomial time" (p. 75).

[15] We follow the established convention in computational complexity theory of setting problem names in small capitals.

[16] It is in fact **PSPACE**-complete.

analyzed the inherent computational complexity of some linguistic problems and came to the conclusion that natural language computations are **NP**-complete. The relevance of Ristad's results, regardless now of their accuracy, is that he derived them independently of any consideration concerning specific linguistic theories, performance models, and so on, which, in his opinion, legitimizes his claim that "[t]he upper and lower bounds of our proposed complexity thesis are tight enough to tell us exactly where the adequate linguistic theories are, not only where they are not" (Ristad 1993: 14). Thus, Ristad takes his results as something with strong implications at the time of building competence/performance models, in the sense that these models will have to be accommodated to the inherent **NP**-completeness of natural language.

Indeed, according to Ristad, the fact that language is in **NP** runs against the modularity thesis, as he interprets the existence of the certificate as the demonstration that the computational system subserving language has access to external information available to verify the correctness of the computations. This is not a proof, of course, but it is a good example of how mathematical results may have some bearing on hypotheses in cognitive science and may help to articulate them, something, by the way, that Lobina sees with a big dose of skepticism when he writes (p. 74) that "the computational complexity of natural language processing will have to consider properties of human psychology such as memory limitations, the strategies that are employed in parsing, the use of the immediate context and many other factors. All in all, it is simply not known what computational complexity our mental machinery exhibits in the processing of language". We, on the other hand, with Ristad, rather believe that it is the task of psycholinguists to incorporate these results when building their performance models, that whatever memory limitations they postulate, whatever parsing strategies they propose, etc., should take into account the inherent structural/computational complexity of natural language. It is perhaps a matter of epistemological priority — what should come first? — and, certainly, a debatable one, but nothing in Lobina's piece actually suggests that he has even the hint of an argument against this idea nor against the idea that this very same strategy can be fruitfully applied to other domains of cognition.

Let's turn to knots, then. To start with, remember that our hypothesis that human knotting abilities might be 'parasitic' on linguistic abilities must be interpreted in the context of an architectural model for (some areas of) the mind according to which a single NCS may underlay more than one cognitive ability. The term 'parasitic' is therefore to be interpreted in the sense of 'sharing computational resources', with such resources being a NCS with specific computational properties. Whence our proposal that, if the complexity of the cognitive tasks associated to knotting abilities were equivalent (or higher, we cannot discard this possibility *a priori*) to that of linguistic abilities, this could constitute an indirect datum at the time of reading the archaeological record, which, so far, shows a strong correlation between the presence of language and knotting. In our attempt to ground on a more formal basis Camps & Uriagereka's (2006) original insights in this connection, we turned our attention to topology and the mathematical

theory of Knots.[17] Lobina finds this outrageous mostly because "the knots that [Knot Theory] studies have nothing to do with real knots" (p. 74), to which it should be added our presumed misinterpretation of a number of complexity results in Knot Theory. We'll discuss these in turn.

Leaving aside Lobina's bold statement, we would like to note first that Knot Theory is not as disconnected from 'reality' (whatever that means) as Lobina wants us to believe it to be. To be sure, Knot Theory was originally motivated to solve a number of problems in organic chemistry and it has thereafter found a number of other applications in several areas of biochemistry and physics; see Adams (2004) for some historical background and examples of these applications. Obviously, that Knots find their application in organic chemistry, for example, is not a demonstration that a relation also exists with the knots one uses when sailing, fishing, mountain climbing, or building a hut. We could cut that story short by just referring to the acknowledgements footnote in the opening of this paper, but we'd rather dwell on this a bit more in order to try to unearth the reasons behind Lobina's skepticism.

One such reason could be the fact that a Knot is defined as an embedding of a circle (S^1) into a sphere (S^3) — denoted $S^1 \rightarrow S^3$ — or, alternatively, as an embedding of S^1 in Euclidean space $\mathbb{R}^3$. That much has the net effect that a Knot must be visualized as a closed structure and not as an open one, as it would be the case, for example, with a sailing line tied to get a bowline. Note, however, that this definition is adopted essentially for practical purposes, since it ensures that Knots are finite objects, but, apart from that, nothing prevents us to define a Knot as the embedding of a tangled line whose two ends extend infinitely into $\mathbb{R}^3$ (Jaume Aguadé, p.c.). Well, yes, there's yet another reason: The embedding could not then be in S^3 (the sphere being a surface delimiting an area of space $\mathbb{R}^3$), given the infinite length of the line, and the homeomorphisms defined in order to determine equivalence relations between Knots could not then be defined as $S^3 \rightarrow S^3$ (i.e. relations between embeddings in a sphere), and should be defined differently. That much, in our opinion, should nevertheless not obscure the fact that Knots can legitimately be taken as models for knots. To strengthen this point, consider the following (informal) definition of a Knot in terms of a knot by Adams (2004: 1–2):

> Take a piece of string. Tie a knot in it. Now glue the two ends of the string together to form a knotted loop. The result is a string that has no loose ends and that is truly knotted. Unless we use scissors, there is no way that we can untangle the string.
>   A *knot* is such a knotted loop of string, except that we think of the string as having no thickness, its cross-section being a single point. The knot is then a closed curve in space that does not intersect itself anywhere.

This is a fairly intuitive way of describing a Knot and we invite the reader to follow Adams's recommendation and to tie an overhand knot and close it afterwards (an electrical extension cord is good for the experiment because it can

---

[17]   In the following discussion we will capitalize the word 'knot' when referring to mathematical entities in order to distinguish the Knots of Knot Theory from the knots of sailors, for example.

be plugged to itself in order to close the knot… or is it a Knot?); or try with a bowline (in both its British and Dutch versions) to see that it is equivalent to a Knot with six crossings. Some fiddling will perhaps be needed with the cord, but, with some practice, one will eventually be able to get something that clearly resembles the standard representation of a Knot (Figure 1).
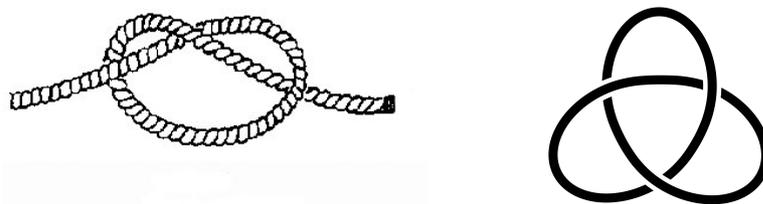

Figure 1: The overhand knot (left) and the trefoil Knot (right). Just close the two loose ends of the knot to get the Knot. (The trefoil image has been generated with the KnotPlot© software developed by Rob Scharein.)

For the sake of completeness, Knots can also be represented in a format that comes closer to knots in the form of braids. A braid is a collection of vertical lines fixed at their two ends to two rigid parallel horizontal bars, such that when they are not tangled each line cuts a horizontal plane parallel to the two bars only once. In fact, a knot more closely resembles a braid consisting of a single knotted line, but it happens that braids are Knot generators, such that when we detach them from the bars and we close them we get a Knot.[18] Braids and Knots are therefore equivalent such that, for example, the trefoil (or threefoil) Knot and its equivalent braid (the overhand knot) can be described by the same braid word σ_1^3 (Adams 2004).

Anyway, a criticism based on the purportedly illegitimate application of the mathematical theory of Knots to some aspect of 'reality', apart perhaps from upsetting a fair number of mathematical realists, strikes us as the same as contending that those areas of physics, chemistry or musical theory that make a fruitful use of group theory are incongruous just because groups are such abstract algebraic structures and so much divorced from reality — what on earth has a group to do with the structure of crystals? That's outrageous!

So much for the question of legitimacy. Of course, from the fact that Knots are perfectly good models for knots it does not follow that they are good models for *cognitive* representations of knots. This point certainly deserves some attention.

One of the inconveniences one faces when trying to investigate the cognitive abilities involved in the act of tying a knot and in the (creative) act of inventing one[19] is the lack of relevant studies on the topic. So far, the only studies

---

18   Not always, i.e. not any closed braid actually generates a Knot, but the reverse is true: Any Knot is equivalent to a closed braid.

19   Lobina (2012: 10) accepts the possibility that "many knots in history came about accidentally". Well, maybe, but not very many of them. Everyone minimally acquainted with knot tying techniques will see this: Perhaps the overhand knot did, but it is highly improbable that the same happened with, for example, the bowline.

we have been able to track down concern experiments with humans or apes (see Michel & Harkins 1985 and Tracy *et al.* 2003 for humans and Herzfeld & Lestel 2005 for apes), focusing on the learning of the relevant motor sequences needed to produce a number of (relatively simple) knots. An interesting aspect of these studies is that in all cases the levels of success are extremely low — for example, in the study by Michel & Harkins (1985) only 37% of the subjects were able to learn to tie the three proposed knots by just attending demonstrations of how to do it, i.e. by just observing the necessary motor sequence to tie them. This, for us, is an indication that knotting abilities have little (if anything) to do with the accurate internalization of a motor sequence. This suspicion is reinforced by the personal experience of some of us with knot tying (and knot learning), since, at least in our cultural tradition, complex knots are taught not through the description of some hand gestures, but rather by resorting to mnemonic techniques whereby the learner is able to figure out *the number and the direction of the crossings* that make up the knot. These, admittedly scant, observations seem to point in one direction, namely that knot production involves at least a particular case of the more general problem of object recognition — i.e. knot recognition, and concomitantly that spatial representation abilities are also involved.

Now, the literature on visual object recognition is abundant, but it is possible to identify an important trend where it is assumed that object recognition involves something akin to parsing in language. In the case of vision, a common view is to assume that object parsing involves the identification of a number of geometric primitives (often cylinders) — this is, for example, the approach of David Marr (1982) or of Irving Biederman (1987), to cite a couple of relevant examples. Underlying this is the assumption that spatial and object representation is entirely based on part-whole, or mereological, relations. This idea, however, has been subjected to several criticisms on the basis that parthood is insufficient to represent and recognize an object and that mereology needs to be complemented with the notion of connectedness, which is eminently topological. Casati & Varzi (1999), for example, have developed a long and detailed argument in favor of the idea that object and spatial representation is mereotopological, not just mereological — a proposal that finds some support in certain experimental results suggesting that object recognition often does not involve parsing (Cave & Kosslyn 1993).

How does all this relate to knots? If in spatial representation some topological relations like connectedness are critically involved, it may well then be the case that our claim that the topological theory of Knots may be relevant is not that far fetched after all. Knots, real ones, have no obvious *parts*, just *crossings* in any of the three spatial dimensions and form a connected (even if open) whole, just like mathematical Knots or braids. Our contention was — and is — that it is this information that is important at the time of producing a knot or figuring out one; in other words that to make a knot, one needs first to represent it and to represent it one needs to figure out its topology. We maintain then that it follows that any act of knot tying, untying, learning, or invention is preceded, minimally by a mental act of knot recognition, which involves representing the basic

properties of the knot, i.e. the number and orientation of its crossings.[20] Fine and accurate motor control will only come afterwards, but, we argue, little headway will be made in, for example, knot learning if the focus is just on hand movements and not on the figure itself. Again, this is a debatable issue, and one can (with Lobina 2012: 10) legitimately stick to the behaviorist assumption that "it is very likely that knot-tying would proceed in a trial-and-error fashion", although, in our opinion, the evidence presented here points in an entirely different direction.

And, so, we have finally come to knot recognition and how it relates to Knot recognition. Let's start with the latter and proceed slowly until we eventually are able to see how it may be associated with the former.

A very important point to be made in order to understand Knot theory's concerns with Knot recognition is condensed in Lobina's following statement: "Knot Theory takes tied knots as its starting assumptions" (p. 75). Which is downright false. Knot theory does not only not take tied Knots as starting assumptions, but it is mainly concerned with proving that there exist other Knots apart from the trivial Knot (the so-called unknot), which is just a closed circle with no crossings. Adams (2004), for example, opens section 1.5 of his book (on page 22) with the following statement: "We have *not yet proved that there is any other knot besides the unknot*. For all we know right now, every projection of a knot […] could simply be a messy projection of the unknot" (emphasis in the original) — despite this being, of course, "the most basic fact of knot theory", i.e. the existence of other Knots apart from the unknot. Just as number theory cannot simply assume the existence of prime numbers — this needs to be proved, Knot theory does not simply assume that there are tied Knots — this also calls for a proof. And here is where Knot recognition comes into play.

From the false statement that Knot theory assumes that there are nontrivial Knots, Lobina derives the false conclusion that it is not the case that one of the fundamental problems in Knot theory is determining whether a string is knotted (p. 75). A couple of quotations will suffice, we hope, to settle this. Here's the first:

> A loop in 3-space, called a *knot*, is unknotted or knotted: it is the fundamental problem of knot theory and we call it *Unknotting*.
> (Hara *et al*. 2005: 359; emphasis in the original)

Another one:[21]

> Determining whether a given knot is trivial or not is one of the historically central questions in topology.   (Agol *et al*. 2005: 3821)

---

[20]   This is not to be interpreted as a commitment with the idea that visual representations of knots are mental images or something similar. The characterization in the text is neutral with respect to that. The important factor is that crossings and orientation are represented somehow, and this can be captured by many different representational formats.

[21]   There is, by the way, a lurking inconsistency in the way Lobina uses the terms 'trivial' and 'non-trivial' when referred to Knots. It seems that, for him, 'trivial' only applies to those projections of the unknot in which it looks like a circle, with 'non-trivial' being applied to those projections in which the unknot appears deformed (see his Figure 1). Another issue that needs to be settled: "The simplest knot of all is just the unknotted circle, which we call the *unknot* or the *trivial knot*" (Adams 2004: 2; emphasis in the original).

And, for good measure, yet another one:[22]

> [W]e will say that links have been *classified* if we can solve the *recognition problem*. That is, is there an algorithm that can decide, in a finite amount of time, if a given pair of links are equivalent? Notice that given such an algorithm, we could then enumerate all links […].
>
> (Hoste 2005: 214; emphasis in the original)

A careful reading of Lobina's text reveals a profound contradiction and, we are afraid, some difficulties in grasping a number of crucial subtleties of Knot theory on his part. Take first his personal account of the UNKNOTTING problem. He writes (p. 74):

> The 'unknotting' problem […] involves specifying an algorithm that can recognize the unknot in a figure like the one found on the right-hand side of Figure 1 (that is, convert the knot on the right-hand side into an unknot).

Firstly, the parenthetical at the end is critical. Note that it presupposes that a deformation of the unknot *is not the unknot*, which it is — the unknot does not cease to be itself in any of its infinitely many different projections (and the Knot appearing on the right-hand side of Figure 1 in Lobina's paper is a projection of the unknot). Secondly, this moreover presupposes that the UNKNOTTING problem just involves transforming a tangled projection of the unknot into an untangled one (a circle) through the successive application of Reidermeister moves, which, yes, are a set of operations that disentangle a Knot without damaging it. But this is *not* the UNKNOTTING problem. It cannot be. The UNKNOTTING problem is *not* an algorithm that can recognize the unknot, but rather an algorithm capable of providing an answer to the question 'is this projection of a Knot a projection of the unknot?' Note that in Lobina's personal interpretation of the problem the answer to this question will always be, trivially, 'yes', because he is assuming that the only projections presented to the algorithm are projections of the unknot and, therefore, the Reidermeister moves will always, sooner or later, convert the projection into a circle.

The point, of course, and this should be clear from the quotations above, is that the algorithm may be presented with a projection of *any* imaginable Knot (the unknot included, of course) and, after the application of the Reidermeister moves, it may turn out that the answer will be 'no'. For this to be possible, however, the algorithm must be able to tell apart the unknot from all other Knots. If it says 'no', the algorithm will certainly not tell us which Knot it is — we will just be certain that it is not the unknot. Note that this is the structure of *any* recognition problem, because recognition (decidability) is defined as the ability to answer 'yes' or 'no' to a specific question concerning some language and, to be able to do that, an algorithm must be equipped with the necessary information to tell apart those elements that belong to the set from those that belong to its complement. When this is not possible, the problem is undecidable, i.e. we may get a 'yes' answer, but we will never get a 'no', with the HALTING problem for

---

[22]   Links are composite Knots, which can be decomposed into 'prime Knots'. Therefore our discussion here focuses only on prime Knots, which are links with only one component.

Turing machines being the model for all undecidable problems.[23] The way Lobina presents UNKNOTTING is as if we understood the problem of recognizing, say, the context-free language $a^n b^n$ as the problem of what a Turing machine (or a push-down automaton for that matter) would do when presented with strings only belonging to that set, which is nonsense.

The reason why UNKNOTTING is so important for Knot theory is because it is precisely the proof that there are other Knots apart from the unknot. It is not an algorithm for listing Knots, of course, nothing like this seems to be forthcoming in the near future, just like no algorithm for calculating all prime numbers will be forthcoming either, although at least we know there are other primes beyond 2.

Finally, note that Lobina's statement that "Knot Theory takes tied knots as its starting assumptions" (which it doesn't) is simply contradictory with the idea that UKNOTTING merely involves disentangling projections of the unknot. If the existence of nontrivial Knots is assumed, why should the task of "working out the formal equivalence of two knots" (p. 74) be just concerned with showing that two projections of the *same* Knot are equivalent? Clearly because UNKNOTTING does not involve working out the formal equivalence of two knots, but rather telling apart the unknot and any Knot projection within the same equivalence class from other Knot projections not in this class. To find a parallelism, UNKNOTTING is as if we were asking 'is x, $x \in \mathbb{Q}$, in the same equivalence class as y, $y \in \mathbb{Q}$?' and we fixed $y = \frac{1}{2}$. Then, with input $x = 2/4$ the answer will be 'yes' and with input $x = 2/3$ the answer will be 'no'. Actually, this formulation of the recognition of equivalence classes in rational numbers comes closer to the other Knot recognition problem analyzed in Hass *et al.* (1999), the paper dealing with the complexity of Knot recognition problems we cited in Balari *et al.* (2011). This problem, which we will refer to as GENUS, is a generalization of UNKNOTTING in the sense that it can be parameterized just like the case of rational equivalence classes by fixing a value for y and then testing whether some x is in the same equivalence class as y. Note that this is precisely the kind of recognition problem to which Hoste (2005) is referring in the quotation above.

To see what GENUS does, we will first have to explain what the genus of a Knot is. Perhaps the best way to do this is with a picture like the one we have in Figure 2.

---

[23]   The reason being, of course, that the complement of HALTING is neither recursive nor recursively enumerable, and decidability implies that if we have a decision algorithm for a set, then we automatically have one for its complement.
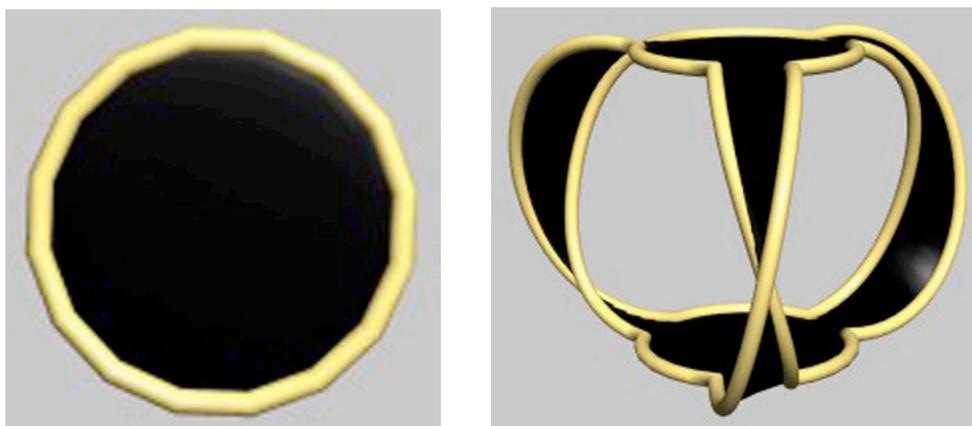
Figure 2: The unknot (left) and the trefoil knot (right) as boundaries of a Seifert surface. The strands of the Knots appear in yellow and the surface is black. (The two images were generated with the SeifertView free software developed by Jarke J. van Wijk, Technische Universiteit Eindhoven.)

As can be seen in the two images of the figure, Knots can be conceptualized as the boundaries of a continuous surface (a Seifert surface) such that the perimeter of the surface is entirely delineated by the strands of the Knot. Observe that in the case of the unknot, the object made up of the Knot plus the surface (Figure 2, left) is something like a drum or a tambourine — the surface has no holes in it. In the case of the trefoil Knot, however, the surface appears perforated (Figure 2, right). It is not immediately obvious from the direct inspection of the image, but the number of holes in the surface is in this case exactly 1.[24] This is the genus of a Knot: The number of holes in the Seifert surface defined by its strands. Note that the genus of the unknot is 0 and that the genus of the trefoil Knot is 1. The genus of a Knot is an indirect indication of its degree of knottedness and, hence, we can formulate a recognition problem in which our question is: 'Is the genus of this Knot equal to (smaller than, greater than) *g*, *g* an integer?' This is the GENUS problem as formulated in Hass *et al*. (1999) and, perhaps more clearly, in Agol *et al*. (2002, 2005). UNKNOTTING is then a particular case of GENUS for the question: 'Is the genus of this Knot equal to 0?'

Now that we have a proper definition of the two Knot recognition problems, we can turn to their complexity results. Hass *et al*. (1999) located UNKNOTTING in **NP** and GENUS in **PSPACE**. This is already an indication that Knot recognition, in its simplest case, is a language (remember that problems are languages) falling within a complexity space closer to that of natural language, since, according to Ristad (1993), NATURAL LANGUAGE is **NP**-complete. It could even be harder, and equivalent to context-sensitive recognition, given the **PSPACE** result for GENUS. In a later study, Agol *et al*. (2002, 2005) reclassified GENUS as **NP**-complete, i.e. exactly in the same complexity class as NATURAL LANGUAGE, while Kuperberg (2011) has recently reclassified UNKNOTTING in **NP** ∩ **coNP**, reducing thus its complexity.

---

[24]   The trick is that it can be demonstrated that the surface is equivalent to a torus (a donut), a surface with a single hole. Other knots will define surfaces equivalent to two or more joined tori, while the unknot defines a surface that is equivalent to a sphere, which has genus 0 (no holes); see Adams (2004: chap. 4).

The question remains open. Newer and more accurate complexity results may be forthcoming for Knot recognition, but it doesn't seem likely that these will locate it in an entirely different complexity space. The computational complexity of natural language and that of Knot recognition are equivalent or very close to each other. These are uncontroversial facts, "not subject to rational debate". What is debatable, of course, is the relevance of Knot recognition complexity results in the assessment of the complexity of knot recognition. It could be the case that knot recognition in humans is a totally different thing from Knot recognition, that it doesn't have anything to do with topology and the identification of crossings in a knot and their orientation, whatever. We are open to discuss this. It's a hypothesis, but a hypothesis that is as informed as it could possibly be given the state of the art of our knowledge of the issues at stake here. Nothing, anyway, that Lobina has been able to really call into question in his paper. And, in the end, it may well be the case that we are on the right track after all.

In his paper, Lobina cites an article by Alan Turing that was unknown to us (Turing 1954), where Turing presents a variety of the Knot recognition problem as one of the challenging puzzles in mathematics. To illustrate the problem, Turing describes a Knot and a method for representing Knots that allows him to reduce a 3D object to a string of symbols. Turing's technique essentially consists of selecting a number of points in a system of Cartesian coordinates and joining them with segments in order to get a closed, connected loop. In addition, Turing encodes the directionality of each segment with a letter, such that a single step from *n* to *n+1* in X is encoded by an **a**, a step from *n* to *n-1* in X is encoded by a **d**, and so on with the other two dimensions, as shown in Figure 3:
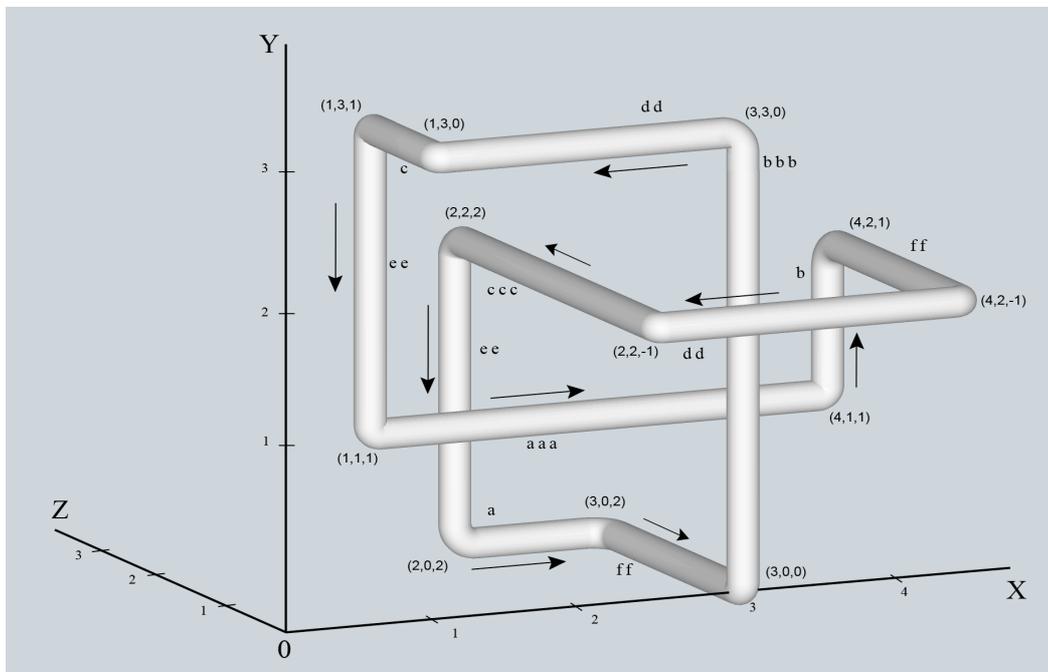

Figure 3: A representation of the trefoil knot in a system of Cartesian coordinates. (Adapted from figure (1b), p. 586, of The Essential Turing, B. Jack Copeland (ed.), Oxford University Press.)

With this representation system, the trefoil Knot can be encoded with the string in (1) and Knot recognition is reduced to a language recognition problem as required by computational complexity theory:

(1) $a_i a_j a_k b_l f_m f_n d_i d_j c_m c_n c_p e_l e_q a_r f_p f_s b_q b_t b_u d_k d_r c_s e_t e_u$

Note that the basic constraint seems to be that the number of as is equal to the number of ds, the number of bs is equal to the number of es, and the number of cs is equal to the number of fs. There are probably others, since it is not clear that any arrangement of these symbols always describes a Knot. Whatever the solution, this mode of representation would open a way to investigate more thoroughly the question of Knot representation and recognition, which as we contended in our earlier papers — and we have come to corroborate here with Turing's system — is not a trivial task. Thus, and depending on what the exact characterization of this 'Knot language' is, we seem to have here a language like the following:

(2) $K = \{a^n c^m e^p b^n d^m f^p\}$

Assuming that the only constraint to adequately describing a Knot is that the number of positive steps in one dimension equals the number of negative steps in the same dimension, but that all positive and negative steps in all three dimensions need not be equal. This is a mildly-context sensitive language with three cross-serial dependencies. However, if the number of positive and negative steps must be the same in all dimensions, then we would seem to have a six-column language like the following:

(3) $K = \{a^n c^n e^n b^n d^n f^n\}$

Which is indexed, and probably equivalent to the triple-copy language in (4), all beyond the power of mildly context-sensitive systems (Radzinski 1991) and, hence possibly residing in a higher complexity class than natural language.

(4) $K = \{www \mid w \in (a+b)^*\}$

Be this as it may, Turing's is a rather clear, and perhaps more intuitive, method for representing Knots that easily captures their two basic properties. Also, it makes their formal complexity more perspicuous, as can be seen from the quick analysis we have just presented.[25] This analysis, even if a rough one, demonstrates that Knots (and knots) are complex objects, but no too complex, perhaps sitting in a region of complexity space similar or not too far away from that of language, as it was originally conjectured by Camps & Uriagereka (2006).

---

[25]   Turing's method is not too different from the Dowker notation for representing Knots, since both give us an indication of the number of crossings. In the Dowker notation each crossing is represented by an ordered pair of integers, *(x, y)*, where *x* is odd and *y* is even, such that the number of pairs (or an ordered list of even numbers) equals the number of crossings in the knot. In this notation, however, it is harder to 'read' the orientation of the crossings than in Turing's, where it can be immediately identified by the change from one symbol to another in the string.

On the whole, and considering the different kinds of data we have presented here, it seems likely that natural computational systems, knots, and language do not define such a bizarre love triangle after all as pretended by Lobina. Paraphrasing Bernard Sumner *et al.*'s (1986) lyrics, we still feel compelled to sing: "If you say the words that I can't say, maybe it's because you can tie the knots that I can't tie". Why should we refrain from investigating it?

Camps & Uriagereka (2006) opened their conclusions section with the following words (p. 63; our emphasis):

> Our conclusion within this paper is not subject to rational debate: knots are *not describable* by any generative procedure that does not have enough operational memory to count as context-sensitive […].

Meaning, of course, that the inherent complexity of knots is an uncontroversial fact. It's there. And any attempt to model human knot-tying abilities will have to take it into account. Just like the results concerning the inherent structural and computational complexity of natural language are there and must serve to drive our research in biolinguistics. This was Camps & Uriagereka's (2006) message and it is our message here.

A fragment taken from the quotation above appears to be one of Lobina's favorites, especially when dropped here and there out of context, or even in the wrong context. It is however surprising to what an extent has Lobina come to believe his own interpretation of the fragment and how faithfully he applies it in his criticisms. A theory, like a knot, is a difficult thing to construct — and like a knot is easier to cut than to entangle, a theory too is easier to trash than to refute. Some people spend their lives telling others that "they don't understand". It would be useful to see what it is that the critics understand, and how that understanding has solutions of any sort to the real problems that science poses, or even how these dynamics allow us to turn absolute mysteries into workable problems. Nothing in Lobina advances our understanding of the problems that were at stake here. It is a classic instance of formal bullying, whereby tools that ought to help us gain insight over our subject matter manage to turn into rhetorical cilices for no discernible purpose other than posturing. It is sad to see how this sort of sophistic logic is often sold as sophisticated reasoning.

References

Adams, Colin. 2004. *The Knot Book. An Elementary Introduction to the Mathematical Theory of Knots*. Providence, RI: American Mathematical Society.

Agol, Ian, Joel Hass & William Thurston. 2002. 3-manifold knot genus is **NP**-complete. In *Proceedings of the Thirty-fourth Annual ACM Symposium on Theory of Computing*, 761–766. New York: ACM Press.

Agol, Ian, Joel Hass & William Thurston. 2005. The computational complexity of knot genus and spanning area. *Transactions of the American Mathematical Society* 358, 3821–3850.

Alle, Henrik & Jörg R. P. Geiger. 2006. Combined analog and action potential coding in hyppocampal mossy fibers. *Science* 311, 1290–1293.

Baddeley, Alan. 2007. *Working Memory, Thought, and Action*. Oxford: Oxford University Press.

Balari, Sergio, Antonio Benítez-Burraco, Marta Camps, Víctor M. Longa & Guillermo Lorenzo. 2010. La importancia de ser moderno. Problemas de método e ideología en el debate sobra la cognición y la conducta de los neandertales. *Ludus Vitalis* XVIII(34), 143–170.

Balari, Sergio, Antonio Benítez-Burraco, Marta Camps, Víctor M. Longa, Guillermo Lorenzo & Juan Uriagereka. 2008. ¿Homo loquens neanderthalensis? En torno a las capacidades simbólicas y lingüísticas del neandertal. *Munibe Antropologia–Arkeologia* 59, 3–24.

Balari, Sergio, Antonio Benítez-Burraco, Marta Camps, Víctor M. Longa, Guillermo Lorenzo & Juan Uriagereka. 2011. The archaeological record speaks: Bridging anthropology and linguistics. *International Journal of Evolutionary Biology* 2011, doi:10.4061/2011/382679.

Balari, Sergio & Guillermo Lorenzo. 2008. Pere Alberch's developmental morphospaces and the evolution of cognition. *Biological Theory* 3, 297–304.

Balari, Sergio & Guillermo Lorenzo. 2009. Computational phenotypes: Where the theory of computation meets Evo-Devo. *Biolinguistics* 3, 2–60.

Balari, Sergio & Guillermo Lorenzo. To appear. *Computational Phenotypes. Towards an Evolutionary Developmental Biolinguistics* (Oxford Studies in Biolinguistics). Oxford: Oxford University Press.

Benítez-Burraco, Antonio, Víctor M. Longa, Guillermo Lorenzo & Juan Uriagereka. 2008. Also sprach Neanderthalis... or did she?. *Biolinguistics* 2, 225–232.

Biederman, Irving. 1987. Recognition-by-Components: A theory of human image understanding. *Psychological Review* 94, 115–147.

Brentari Diane (ed.) 2010. *Sign Languages*. New York: Cambridge University Press.

Camps, Marta & Juan Uriagereka. 2006. The Gordian Knot of linguistic fossils. In Joana Rosselló & Jesús Martín (eds.), *The Biolinguistic Turn. Issues on Language and Biology*, 34–65. Barcelona: PPU.

Casati, Roberto & Achille C. Varzi. 1999. *Parts and Places. The Structures of Spatial Representation*. Cambridge, MA: MIT Press.

Cave, Carolyn Baker & Stephen M. Kosslyn. 1993. The role of parts and spatial relations in object identification. *Perception* 22, 229–248.

Cheng, Hsiang-Chun, Hung-Yi Chen, Chia-Liang Tsai, Yung-Jung Chen & Ron-Ju Cherng. 2009. Comorbidity of motor and language impairments in preschool children of Taiwan. *Research in Developmental Disabilities* 30, 1054–1061.

Chomsky, Noam. 1956. On the limits of finite-state description. *Quarterly Progress Report* 48, 116–118.

Chomsky, Noam. 1957. *Syntactic Structures*. The Hague: Mouton.

Chomsky, Noam. 1959. On certain formal properties of grammars. *Information and Control* 2, 137–168.

Chomsky, Noam. 1963. Formal properties of grammars. In R. Duncan Luce, Robert R. Bush & Eugene Galanter (eds.), *Handbook of Mathematical Psychology*, vol. II, 323–418. New York, NY: John Wiley.

Chomsky, Noam. 2007. Approaching UG from below. In Uli Sauerland & Hans-Martin Gärtner (eds.), *Interfaces + Recursion = Language? Chomsky's Minimalism and the View from Syntax–Semantics*, 1–29. Berlin: de Gruyter.

Chomsky, Noam. 2010. Some simple evo devo theses: How true might they be for language. In Richard K. Larson, Viviane Déprez & Hiroko Yamakido (eds.), *The Evolution of Language: Biolinguistic Perspectives*, 45–62. Cambridge: Cambridge University Press.

Coolidge, Frederick L. & Thomas Wynn. 2004. A cognitive and neuropsychological perspective on the Châtelperronian. *Journal of Anthropological Research* 60, 55–73.

Del Vecchio, Domitilla, Richard M. Murray & Pietro Perona. 2003. Decomposition of human motion into dynamics-based primitives with application to drawing tasks. *Automatica* 39, 2085–2098.

Devitt, Michael. 2006. *Ignorance of Language*. Oxford: Oxford University Press.

Dipietro, Laura, Hermano I. Krebs, Susan E. Fasoli, Bruce T. Volpe & Neville Hogan. 2009. Submovement changes characterize generalization of motor recovery after stroke. *Cortex* 45, 318–324.

Earley, Jay. 1970. An efficient context-free parsing algorithm. *Communications of the ACM* 13, 94–102.

Eden, Guinevere F., Frank B. Wood & John F. Stein. 2003. Clock drawing in developmental dyslexia. *Journal of Learning Disabilities* 36, 216–228.

Fisher, Simon & Constance Scharff. 2009. *FOXP2* as a molecular window into speech and language. *Trends in Genetics* 25, 166–177.

Fitch, W. Tecumseh & Marc D. Hauser. 2004. Computational constraints on syntactic processing in nonhuman primates. *Science* 303, 377–380.

Flash, Tamar & Binyamin Hochner. 2005. Motor primitives in vertebrates and invertebrates. *Current Opinion in Neurobiology* 15, 660–666.

Fodor, Jerry A. 1975. *The Language of Thought*. New York: Thomas & Crowell.

Fodor, Jerry A. & Massimo Piattelli-Palmarini. 2010. *What Darwin Got Wrong*. New York: Farrar, Straus and Giroux.

Fujita, Eriko, Yuko Tanabe, Akira Shiota, Masatsugu Ueda, Kiyotaka Suwa, Mariko Y. Momoi & Takashi Momoi. 2008. Ultrasonic vocalization impairment of *Foxp2* (R552H) knockin mice related to speech-language disorder and abnormality of Purkinje cells. *Proceedings of the National Academy of Sciences USA* 105, 3117–3122.

Fung, Victor S., Morris, John G., Jonathan Leicester, Y. S. Soo & Llewelyn Davies. 1997. Clonic perseveration following thalamofrontal disconnection: A distinctive movement disorder. *Movement Disorders* 12, 378–385.

Gentner, Timothy Q., Kimberly M. Fenn, Daniel Margoliash & Howard C. Nusbaum. 2006. Recursive syntactic pattern learning in songbirds. *Nature* 440, 1204–1207.

Gopnik, Myrna. 1990. Feature-blind grammar and dysphasia. *Nature* 344: 715.

Grossman, Murray G., S. Carvell, S. Gollomp, M. B. Stern, G. Vernon & H. I. Hurtig. 1991. Sentence comprehension and praxis deficits in Parkinson's disease. *Neurology* 41, 1620–1628.

Gusella, James F. & Marcy E. MacDonald. 2006. Huntington's disease: seeing the pathogenic process through a genetic lens. *Trends in Biochemical Sciences* 31,

533–540.

Hara, Masao, Seiichi Tani & Makoto Yamamoto. 2005. UNKNOTTING is in **AM** ∩ **co-AM**. *16th ACM-SIAM Symposium on Discrete Algorithms* (SODA 2005), 359–364, Vancouver, BC.

Hass, Joel, Jeffrey C. Lagarias & Nicholas Pippenger. 1999. The computational complexity of knots and link problems. *Journal of the ACM* 46, 185–211.

Hauser, Marc D., Noam Chomsky & W. Tecumseh Fitch. 2002. The faculty of language: What is it, who has it, and how did it evolve? *Science* 298, 1569–1579.

Heijningen, Caroline A. A. van, Jos de Visser, Willem Zuidema & Carel ten Cate. 2009. Simple rules can explain discrimination of putative recursive syntactic structures by a songbird species. *Proceedings of the National Academy of Sciences USA* 106, 20538–20543.

Herzfeld, Chris & Dominique Lestel. 2005. Knot tying in great apes: Etho-ethnology of an unusual tool behavior. *Social Science Information* 44, 621–653.

Hinzen, Wolfram. 2006. *Mind Design and Minimal Syntax*. Oxford: Oxford University Press.

Hopcroft, John E. & Jeffrey D. Ullman. 1979. *Introduction to Automata Theory, Languages, and Computation*. Reading, MA: Addison-Wesley.

Hoste, Jim. 2005. The enumeration and classification of knots and links. In William W. Menasco & Morwen B. Thistlethwaite (eds.), *The Handbook of Knot Theory*, 209–232. Amsterdam: Elsevier.

Hudson, Kerry D. & Emily K. Farran. 2010. Drawing the line: Drawing and construction strategies for simple and complex figures in Williams syndrome and typical development. *British Journal of Developmental Psychology* 29, 687–706.

Iverson, Jana M. & Barbara A. Braddock. 2011. Gesture and motor skill in relation to language in children with language impairment. *Journal of Speech, Language, and Hearing Research* 54, 72–86.

Karmiloff-Smith, Annette. 2006. Williams Syndrome. In Keith Brown (ed.), *Encyclopedia of Language and Linguistics*, vol. 13, 585–589, Oxford: Elsevier.

Khabbaz, Nabil A. 1974. A geometric hierarchy of languages. *Journal of Computer and System Sciences* 8, 142–157.

Kuperberg, Greg. 2011. Knottedness is in **NP**, modulo GRH. [http://arxiv.org/abs/1112.0845] (4 February 2012).

Lewis, Harry R. & Christos H. Papadimitriou. 1981. *Elements of the Theory of Computation*. Englewood Cliffs, NJ: Prentice-Hall.

Lieberman, Philip. 2000. *Human Language and Our Reptilian Brain. The Subcortical Bases of Speech, Syntax and Thought.* Cambridge, MA: Harvard University Press.

Lieberman, Philip. 2002. On the nature and evolution of the neural bases of human language. *American Journal of Physical Anthropology* 45, 36–62.

Lieberman, Philip. 2006. *Toward an Evolutionary Biology of Language*. Cambridge, MA: Harvard University Press.

Liégeois, Frédérique, Torsten Baldeweg, Alan Connelly, David G. Gadian, Mortimer Mishkin & Faraneh Vargha-Khadem. 2003. Language fMRI ab-

normalities associated with *FOXP2* gene mutation. *Nature Neuroscience* 6, 1230–1237.

Lipowska, Malgorzata, Ewa Czaplewska & Anna Wysocka. 2011. Visuospatial deficits of dyslexic children. *Medical Science Monitor* 17, CR216–221.

Lobina, David. 2012. Conceptual structure and the emergence of the language faculty: Much ado about knotting. Ms., Tarragona: Universitat Rovira i Virgili. *LingBuzz*, http://ling.auf.net/lingBuzz/001397.

MacSweeney, Mairéad, Cheryl M. Capek, Ruth Campbell & Bencie Woll. 2008. The signing brain: the neurobiology of sign language. *Trends in Cognitive Sciences* 12, 432–440.

Makuuchi, Michiru. 2010. fMRI studies on drawing revealed two new neural correlates that coincide with the language network. *Cortex* 46, 268–269.

Makuuchi, Michiru, Tatsuro Kaminaga & Morihiro Sugishita. 2003. Both parietal lobes are involved in drawing: a functional MRI study and implications for constructional apraxia. *Cognitive Brain Research* 16, 338–347.

Marcus, Gary & Simon E. Fisher. 2003. *FOXP2* in focus: what can genes tell us about speech and language? *Trends in Cognitive Sciences* 7, 257–262.

Marr, David. 1982. *Vision*. San Francisco, CA: Freeman.

Michel, George F. & Debra A. Harkins. 1985. Concordance of handedness between teacher and student facilitates learning manual skills. *Journal of Human Evolution* 14, 597–601.

Minelli, Alessandro. 2011. A principle of developmental inertia. In Benedikt Hallgrímsson & Brian K. Hall (eds.), *Epigenetics. Linking Genotype and Phenotype in Development and Evolution*, 116–133. Berkeley, CA: University of California Press.

Murray, Laura L. 2000. Spoken language production in Huntington's and Parkinson's diseases. *Journal of Speech, Language, and Hearing Research* 43, 1350–1366.

O'Donnell, Timothy J., Marc D. Hauser & W. Tecumseh Fitch. 2005. Using mathematical models of language experimentally. *Trends in Cognitive Sciences* 9, 284–289.

Papadimitriou, Christos H. 1994. *Computational Complexity*. Reading, MA: Addison-Wesley.

Pavlidou, Elpis V., M. Louise Kelly & Joanne M. Williams. 2010. Do children with developmental dyslexia have impairments in implicit learning? *Dyslexia* 16, 143–161.

Perruchet, Pierre & Arnaud Rey. 2005. Does the mastery of center-embedded linguistic structures distinguish humans from non-humans?. *Psychonomic Bulletin and Review* 12, 307–313.

Petersson, Karl Magnus, Vasiliki Folia & Peter Hagoort. 2010. What artificial grammar learning reveals about the neurobiology of syntax. *Brain and Language*, doi 10.1016/j.bandl.2010.08.003.

Radzinski, Daniel. 1991. Chinese number-names, Tree Adjoining Languages, and mild context-sensitivity. *Computational Linguistics* 17, 277–299.

Rechetnikov, Rouslan P. & Kinsuk Maitra. 2009. Motor impairments in children associated with impairments of speech or language: A meta-analytic review of research literature. *The American Journal of Occupational Therapy* 63, 255–

263.

Ristad, Eric Sven. 1993. *The Language Complexity Game*. Cambridge, MA: The MIT Press.

Rogers, James & Marc D. Hauser. 2010. The use of formal language theory in studies of artificial language learning: A proposal for distinguishing the differences between human and nonhuman animal learners. In Harry van der Hulst (ed.), *Recursion and Human Language*, 213–232. Amsterdam: Mouton de Gruyter.

Rogers, James & Geoffrey K. Pullum. 2011. Aural pattern recognition experiments and the Subregular Hierarchy. *Journal of Logic, Language and Information* 20, 329–342.

Sarpeshkar, Rahul. 1998. Analog versus digital: Extrapolating from electronics to neurobiology. *Neural Computation* 10, 1601–1638.

Sarpeshkar, Rahul. 2009. Neuromorphic and biomorphic engineering systems. In *McGraw-Hill Yearbook of Science & Technology 2009*, 250–252. New York: McGraw-Hill.

Shriberg, Lawrence D., Kirrie J. Ballard, J. Bruce Tomblin, Joseph R. Duffy, Katharine H. Odell & Charles A. Williams. 2006. Speech, prosody, and voice characteristics of a mother and daughter with a 7,13 translocation affecting *FOXP2*. *Journal of Speech, Language, and Hearing Research* 49, 500–525.

Shu, Yousheng, Andrea Hasenstaub, Álvaro Duque, Yuguo Yo & David Mc-Cormick. 2006. Modulation of intracortical synaptic potentials by presynaptic somatic membrane potential. *Nature* 441, 761–765.

Shu, Weiguo, Julie Y. Cho, Yuhui Jiang, Minhua Zhang, Donald Weisz, Gregory A. Elder, James Schmeidler, Rita De Gasperi, Miguel A. Gama Sosa, Donald Rabidou, Anthony C. Santucci, Daniel Perl, Edward Morrisey & Joseph D. Buxbaum. 2005. Altered ultrasonic vocalization in mice with a disruption in the *Foxp2* gene. *Proceedings of the National Academy of Sciences USA* 102, 9643–9648.

Striedter, Georg F. 2005. *Principles of Brain Evolution*. Sunderland, MA: Sinauer.

Sumner, Bernard, Gillian Gilbert, Peter Hook & Stephen Morris. 1986. Bizarre love triangle. In New Order, *Brotherhood* (FACT 150), side two, track 1. Manchester: Factory.

Surmeier, D. James, Jaime N. Guzmán, Javier Sánchez-Padilla & Joshua A. Goldberg. 2010. What causes the death of dopaminergic neurons in Parkinson's disease? *Progress in Brain Research* 183, 59–77.

Tomalin, Marcus. 2007. Reconsidering recursion in syntactic theory. *Lingua* 117, 1784–1800.

Tomalin, Marcus. 2011. Syntactic structures and recursive devices: A legacy of imprecision. *Journal of Logic, Language and Information* 20, 297–315.

Tracy, Joseph, Adam Flandrers, Saussan Madi, Joseph Laskas, Eve Stoddard, Ayis Pyrros, Peter Natale & Nicole DelVecchio. 2003. Regional brain activation associated with different performance patterns during learning of a complex motor skill. *Cerebral Cortex* 13, 904–910.

Troyer, Angela K., Sandra E. Black, Maria L. Armilio & Morris Moscovitch. 2004. Cognitive and motor functioning in a patient with selective infarction of the

left basal ganglia: evidence for decreased non-routine response selection and performance. *Neuropsychologia* 42, 902–911.

Turing, Alan M. 1954. Solvable and unsolvable problems. In B. Jack Copeland (ed.), *The Essential Turing*, 576–595. Oxford: Oxford University Press.

Ullman, Michael T. 2001. The declarative/procedural model of lexicon and grammar. *Journal of Psycholinguistic Research* 30, 37–69.

Ullman, Michael T. 2004. Contributions of memory circuits to language: the declarative/procedural model. *Cognition* 92, 231–270.

Vargha-Khadem, Faraneh, Kate E. Watkins, Katie J. Alcock, Paul Fletcher & Richard E. Passingham. 1995. Praxic and nonverbal cognitive deficits in a large family with a genetically transmitted speech and language disorder. *Proceedings of the National Academy of Sciences USA* 92, 930–933.

Vargha-Khadem, Faraneh, Kate E. Watkins, C. J. Price, John Ashburner, Katie J. Alcock, Alan Connelly, Richard S. J. Frackowiak, Karl J. Friston, M. E. Pembrey, Mortimer Mishkin David G. Gadian & Richard E. Passingham. 1998. Neural basis of an inherited speech and language disorder. *Proceedings of the National Academy of Sciences USA* 95, 12695–12700.

Vargha-Khadem, Faraneh., David G. Gadian, Andrew Copp & Mortimer Mishkin. 2005. *FOXP2* and the neuroanatomy of speech and language. *Nature Reviews Neuroscience* 6, 131–138.

Vicari, Stefano, A. Finzi, Deny Menghini, Luigi Marotta, S. Baldi & Laura Petrosini. 2005. Do children with developmental dyslexia have an implicit learning deficit? *Journal of Neurology, Neurosurgery & Psychiatry* 76, 1392–1397.

Vijay-Shanker, K. & David J. Weir. 1994. The equivalence of four extensions of context-free grammars. *Mathematical Systems Theory* 27, 511–546.

Waddington, Conrad H. 1957. *The Strategy of the Genes*. London: Allen and Unwin.

Watkins, Kate E., Faraneh Vargha-Khadem, John Ashburner, Richard E. Passingham, Alan Connelly, Karl J. Friston, Richard S. J. Frackowiak, Mortimer Miskin & David G. Gadian. 2002a. MRI analysis of an inherited speech and language disorder: structural brain abnormalities. *Brain* 125, 465–478.

Watkins, Kate E., Nina F. Dronkers & Faraneh Vargha-Khadem. 2002b. Behavioural analysis of an inherited speech and language disorder: Comparison with acquired aphasia. *Brain* 125, 452–464.

Webster, Richard I., Annette Majnemer, Robert W. Platt & Michael I. Shevell. 2005. Motor function at school age in children with a preschool diagnosis of developmental language impairment. *Journal of Pediatrics* 146, 80–85.

Weir, David J. 1992. A geometric hierarchy beyond context-free languages. *Theoretical Computer Science* 104, 235–261.

Weir, David J. 1994. Linear iterated pushdowns. *Computational Intelligence* 10, 431–439.

Williams, George C. 1966. *Adaptation and Natural Selection*. Oxford: Oxford University Press.

Wynn, Thomas & Frederick L. Coolidge. 2011. The implications of the working memory model for the evolution of modern cognition. *International Journal of Evolutionary Biology*, doi:10.4061/2011/741357.

Zylberberg, Ariel, Stanislas Dehaene, Pieter R. Roelfsema & Mariano Sigman. 2011. The human Turing machine: A neural framework for mental programs. *Trends in Cognitive Sciences* 15, 293–300.

Sergio Balari
Universitat Autònoma de Barcelona
& Centre de Lingüística Teòrica
Departament de Filologia Catalana
Edifici B, Campus UAB
08193 Bellaterra (Barcelona)
Spain
Sergi.Balari@uab.cat

Antonio Benítez-Burraco
Universidad de Huelva
Departamento de Filología Española
*   y sus Didácticas*
Campus El Carmen
21071 Huelva
Spain
antonio.benitez@dfesp.uhu.es

Marta Camps
George Washington University
Department of Anthropology
Center for the Advanced
Study of Human Paleobiology
Washington DC 20052
USA
mcamps@gwu.es

Víctor M. Longa
Universidade de Santiago de Compostela
Departamento de Literatura Española, Teoría
da Literatura e Lingüística Xeral
Campus Norte
15782 Santiago de Compostela
Spain
victormanuel.longa@usc.es

Guillermo Lorenzo
Universidad de Oviedo
Departamento de Filología Española
Facultad de Filosofía y Letras, Campus El Milán
33011 Oviedo
Spain
glorenzo@uniovi.es