Linguistic scaffolding and the epistemic utility of language-like representation

Whit Schonbein

Department of Computer Science
University of New Mexico, Albuquerque

whit.schonbein at gmail dot com

DRAFT SEPTEMBER 2013

ABSTRACT: The thesis of linguistic scaffolding holds that extra-cranial linguistic entities – utterances or written sentences – can redirect or transform on-board cognitive resources in such a way that agents interacting with those entities are able to perform tasks they would otherwise not be able to accomplish. Proponents of scaffolding typically also have an aversion to linguistically-structured mental representation: Scaffolding not only enables new forms of thought, but leaves combinatorial structure in the world. I argue that this strategy is epistemically unfortunate, because taking the linguistic structure of mental representation seriously can provide valuable insight into why linguistic scaffolding occurs, and what it is doing when it does.

1. Introduction

The thesis of linguistic scaffolding holds that in addition to their communicative roles, extra-cranial linguistic entities – utterances, written sentences – can redirect or transform on-board cognitive resources in such a way that agents interacting with those entities are able to perform tasks they would otherwise not be able to accomplish. Proponents of scaffolding typically also have a strong aversion to linguistically-structured mental representation, one seemingly inherited from prior arguments between proponents of the 'language of thought' and advocates of a putative 'connectionist' or artificial neural network alternative (Fodor 1975, 1987; Fodor & Pylyshyn 1988; Clark 1993). Combining these two positions, the rough idea is that combinatorial or constituent linguistic structure can be left in the extra-cranial environment, where it *scaffolds* non-linguistic, on-board representational schemes (e.g., vectors, points in state space, non-propositional, subsymbolic, etc.), thereby enabling new forms of behavior.

The reluctance to allowing linguistic structure in the head is both problematic and unfortunate. It is problematic because, as others have observed, some aspects of linguistic scaffolding may actually imply that linguistic structure is internally recapitulated (Wheeler 2004, 2007; Schonbein 2012). More broadly, an alternative tradition concerning representation in 'non-classical' automata holds that such representations are just as 'language-like' as schemes used in automata such as Turing machines; the issue is not whether on-board representations are language-like, but rather what the language is and how it is implemented. The reluctance is unfortunate because recognizing that on-board representational schemes are linguistically structured can help us understand why linguistic scaffolding occurs, and what it is doing when it does.

The primary goal of this paper is to motivate the second of these claims, i.e., that taking the linguistic structure of mental representation seriously can aid in our understanding of scaffolding. In section 2, I explore the linguistic scaffolding thesis in more detail, and summarize how it conflicts with the desire to keep compositional structure outside the head; I do not attempt an exhaustive defense of these arguments. Rather, in section 3, working under the assumption that it is reasonable to posit linguistically-structured mental representations, I provide two examples of how positing such structure facilitates our understanding of scaffolding. Finally, I conclude by generalizing the results of sections 1 and 2 by relating them to Kirsh's (1990) and Clark's (1992) discussion of explicit and implicit information.

2. Scaffolding and Linguistic Structure

Human infants, when exposed to instances drawn from a set of images (e.g., cartoon faces), form a visual category with the average of those instances serving as a prototype. Plunkett et. al. (2008) divide the same set of images into two exclusive subsets, and each subset is given a nonsense label, so that during the exposure phase infants are shown both an image and its label. In this case, infants form two visual categories rather than one, with the within-group average serving as the prototype for each. This is a lucid example of how external language can shape cognitive resources into 'otherwise unattainable' configurations by 'diverting' – perhaps through attentional mechanisms – the on-board representational processes from their default course of action. This is just one of many putative examples of scaffolding, including but not limited to connectionist networks performing long multiplication or logical proof (Rumelhart, et. al. 1986, Bechtel & Abrahamsen 2002), ship navigation (Hutchins 1995), nest building by termites (Clark 1997), path finding using external markers in Argentinian ants (Garnier et. al. 2013), same-different object categorization by primates (Clark 1998, 2008), and numerical reasoning (Dahaene 1997, Clark 2008).

Within these examples one finds two non-exclusive 'flavors' of scaffolding: *On-line* and *internalized*. In on-line scaffolding, problems are solved through the real-time interaction between external structures and on-board processes (e.g., visual category formation in infants).[1] In other words, the external entities are essential to task completion in the sense they provide a concurrent means for overcoming limitations of on-board processes (or, alternatively, on-board processes are 'configured' to couple with those external structures so as to solve the problem). In this case, there is no expectation that the agent could solve the problem without the assistance of those structures.

In contrast, internalized scaffolding occurs when external entities provide a source of structure that is 'absorbed' by on-board cognitive systems.[2] The process of internalizing external structure modifies on-board processes so that they can serve as surrogates in situations when the external props are not available. For instance, after speculating that connectionist networks supplemented with external arrays of numerals could perform arbitrary multiplication problems, Rumelhart et. al. suggest that these numerical arrays could be internalized into the network's own representational processes:

> Not only can we manipulate the physical environment and then process it, we can also learn to internalize the representations we create, "imagine" them, and then process these imagined representations – just as if they were external. (p. 46)

As a result, we can multiply without the aid of external symbols, i.e., "[w]e can … imagine writing

1 Cf. Wheeler & Clark's (1999) distinction between 'on-line' and 'off-line' intelligence.
2 Proponents of scaffolding tend to avoid equating internalization with learning, so here I follow suit.

down a multiplication problem and imagine multiplying them together" (p. 46). External linguistic structures cause internal restructuring, resulting in the ability to perform tasks previously impossible without the aid of external resources.[3,4]

Internalization plays a central role in some theorists' accounts of more complex mental processes. For example, Dennett's multiple-drafts account of consciousenss is based upon the speculation that language installs a 'virtual serial machine' over the massively parallel architecture of the non-linguistic brain (Dennett, 1991). Similarly, Clark's account of "second-order cognitive dynamics", i.e., the capacity to "[think] about our own cognitive profiles or specific thoughts" (1998, p. 177), appeals to internalization. Roughly speaking, Clark's proposal is that exposure to public language allows us to treat our own thoughts, as expressed in these sentences, as objects of attention in their own right (hence the 'second-order'). Furthermore, these external linguistic entities are internalized in the sense that what used to be extra-cranial can now be deployed internally, as a form of 'inner rehearsal' or 'inner speech', resulting in the ability to reflect on our own mental states in the absence of external scaffolding (Clark 1998, 2008). Internalization is thus the key to unlocking much of a human's mental life.

On-line linguistic scaffolding is typically interpreted as facilitating the parallel yet independent goal of minimizing the complexity of on-board resources by leaving such complexity in the extra-cranial environment. Specifically, proponents often resist the view that mental representations comprise a 'language of thought' exhibiting a compositional structure akin to that of phrases and sentences in natural language (Fodor 1975; Fodor & Pylyshyn 1988). Instead of being served by language-like representational processes, proponents claim, the relevant tasks can be solved by alternative strategies when supplemented with appropriate linguistically-structured scaffolding (Clark 1994, 2008; Rowlands 1999, 2009; Rumelart et. al, 1986). For instance, Rumelhart et. al.'s speculative network is explicitly anti-linguistic: On their account, connectionist networks engage in pattern-matching facilitated by putatively non-linguistic representations such as activation vectors, microfeatures, and the like. By supplementing the network with *external* linguistic structures, long multiplication is transformed from a 'language-like', rule-governed symbol manipulation task into a pattern-matching one. A similar position is taken by Clark (1997, 2008), who holds that the 'natural' method of computation for brains is pattern matching.

There is a glaring tension between the goal of externalizing linguistic structure and internalizing scaffolding. On the one hand, internalization involves transforming cognitive processes by somehow bringing external structure on-board. On the other hand, the resulting modifications to those processes

3 One of Clark's (1998, 2008) preferred examples also calls for internalization. Chimpanzees can perform same-different tasks: When given two objects, the animal must identify whether they are instances of the same type (e.g., two cups) or instances of different types (e.g., a cup and a shoe). In contrast, they have difficulty performing the higher order 'conceptual matching' task of determining whether two pairs both instantiate sameness or difference (e.g., a pair of cups and a pair of shoes), or whether one instantiates sameness while the other realizes difference (a pair consisting of a cup and a shoe, and a pair of cups). However, when the initial task is learned with the addition of labels – e.g., plastic tags, where matching pairs are marked with green tags and mismatched pairs are marked with blue tags – the conceptual-matching task can be solved. Importantly, in this later task the tags are not physically present. Rather, according to Clark, their influence is felt through internalization and recall.

4 It's possible (although perhaps not elegant) to interpret the infant visual category results as involving internalization. Perhaps in the case where labels are presented with each image, those tags are internalized as visual or linguistic categories, and then those internal structures are the primary causal factors involved in the formation of the other categories. After all, unlike successive images, which differ from each other, instances of each type of label do not vary across presentations, so it may be that infants quickly form stable on-board categories for those labels relative to categories for the images themselves, and the causal influence of these visio-linguistic categories ripples outwards, influencing the formation of the slower-forming image categories.

are supposed to *avoid* recapitulating the very structure deemed so important to the cognitive capacities it enables, e.g., compositional syntax. How can proponents have it both ways?

The most well-developed response to this question is given by Clark, who proposes that linguistic structure is part of the *content* of on-board representations, but not part of their structure. For example,

> [j]ust as I can represent greenness without deploying a green inner vehicle, so too I can represent a sentence as involving three component ideas (John, loving, and Mary, to stick with the tired old example) without thereby deploying an inner vehicle that itself comprises three distinct symbols exhibiting that articulation. (Clark 2004, p. 722)

In short, representing a specific color does not require representational vehicles be that color, and neither does representing a specific compositional structure require the vehicles to have that structure.

It is not clear that this strategy is successful. For example, Wheeler (2004, 2007) argues that Clark's (1998, 2008) claim that language can be internalized into a form of 'inner speech' implies the recapitulation of at least some linguistic structure by on-board representations. According to Wheeler, if it is the structure of external linguistic entities that transforms cognition, and cases of inner speech are cases where cognition is so transformed in the absence of those external entities, then this is sufficient reason to conclude that the structure must be present in the on-board mechanisms.

In contrast, Schonbein (2012) targets proponents of scaffolding who adopt a broadly connectionist account of mental representation (such as Clark), arguing that research on formal language processing in finite-node recurrent artificial neural networks implies that constituent structure is replicated in on-board representations whenever that structure must be processed by those mechanisms, e.g., in the case making grammatical judgments. This argument is independent of any claims about inner speech or even internalization: It is irrelevant whether the capacity to parse sentences is the result of ‘internalization’ or enables ‘recall’ – if on-board processes are connectionist and the system parses sentences, Schonbein argues, constituent structure is recapitulated.

Finally, it pays to keep in mind that there is an alternative interpretation of networks and other putative non-classical systems available in the literature, one that contrasts with the standard account. According to the standard interpretation, 'non-classical' automata (such as neural networks) represent in a manner very different than ‘classical’ systems do, namely, through vehicles that do not recapitulate linguistic structure. The alternative interpretation – one that has received far less attention in philosophical literature – holds that automata differ not in whether their representations are language-like, but rather in how language-like schemes are *implemented*. For example, on this interpretation it is not the case that connectionist representations lack representations with constituent structure (and rules sensitive to this structure); rather, they simply realize this structure in a different way than prototypical classical systems such as Turing machines or von Neumann architectures.[5]

This interpretation is consistent with how the issue is viewed in computer science. From that perspective, automata such as networks, Turing machines, push-down automata, quantum computers, the Game of Life, aperiodic tiles, cardiac muscle, *Magic: The Gathering*, and other equivalent computational automata differ not in whether they compute languages (they do), but rather in *how* they do it.[6] This is because *all* computational systems fall in the hierarchy of computational complexity, and

5 An example of this interpretation can be found in Aydede (1997), who surveys then-current connectionist models, arguing that each can be interpreted as implementing language-like representational schemes. There are elements of this position in Schonbein (2012), as well.

6 Demonstrations of the Turing-equivalence of different network architectures can be found in Ceisinger (1988),

therefore compute formal languages (Chomsky 1963). This holds even when we allow that a computational system is spatially or temporally continuous; in such cases, either the additional 'super-Turing' power afforded by the introduction of continuity (if there is any) can itself be viewed from the perspective of language computation (i.e., identified with a new class of languages more complex than context-free languages), or, if we bring physical implementation into the picture, the additional power disappears in the noise of the physical world (Bournez & Campagnolo 2008). From the computational perspective, the only way to escape language-like representation is to demonstrate the existence of algorithmic mechanisms that resist assimilation into the mathematics of computation, an accomplishment on par with Gödel's incompleteness proof (after all, such a demonstration would disprove the Church-Turing thesis). From a computational perspective, it is hard to see how appeals to scaffolding can avoid language-like internal representations, because such representations are ubiquitous.[7]

If these concerns are correct, then what is interesting about linguistic scaffolding is the suite of concerns familiar from computer science: The complexity of on-board representational schemes (regular, context-sensitive, context-free, etc.), their structure (states, operations, etc.), how they are implemented (two-dimensional arrays, nested regions of multidimensional state space, quantum properties, etc.), and their efficiency (e.g. polynomial, non-polynomial, etc.). Besides fostering interdisciplinary cohesion, integrating scaffolding into computational theory carries with it certain benefits, such as an ability to ask concise questions about the effects of scaffolding (e.g., In what sense does it decrease the complexity of on-board resources? Can it make problem solving mechanisms more time or space efficient? How well would certain scaffolding strategies scale as the size of the problem increases?). Furthermore, taking seriously the idea that cognitive behavior is the result of implementing language-like representational processes in physical devices can help us predict and explain situations where linguistic scaffolding arises. Specifically, when we know something about the formal (linguistic) structure of a representational mechanism, and we know how that mechanism may be implemented in the head, we gain insight into why and when scaffolding occurs – or so I claim.

Siegelmann & Sontag (1994), and Franklin & Garzon (1991). Moore & Mertens (2011) give an overview of results regarding quantum computation, Berlekamp (1982) demonstrates that the Game of Life is Turing-equivalent, Grunbaum & Shephard (1987) show how aperiodic tilings compute, Scarle (2009) describes how heart tissue is Turing-equivalent, while Churchill (n.d.) demonstrates the Turing-equivalence of the fantasy card game, *Magic: The Gathering*. In short, all of these divers systems share the property of computing languages.

7 Obviously there is a lot more that could be said regarding this argument sketch. One popular reply is that the language of thought appeals to a more substantive notion of 'language-like representation' than the admittedly ubiquitous mathematical concept. Specifically, the language of thought is (i) committed to a certain view about the semantics of mental representation (e.g., Fodor 1987), and (ii) makes substantive claims about how languages are implemented. Regarding (i), while the claim that the proper semantics for mental representation resides at the level of conceptual or symbolic content (rather than 'subconceptual' or 'subsymbolic' content) is extremely interesting, it is independent of the present concerns, since we can equally entertain a 'subsymbolic' semantics for Turing machines or von Neumann architectures as we can for networks, and we can just as easily entertain a 'symbolic' semantics for networks as we can for Turing machines. The choice of symbolic or subsymbolic semantics does not distinguish between types of automata. Regarding (ii), we are told that the language of thought uses representations of the sort we find in Turing or von Neumann machines. However, proponents of the language of thought consistently retreat from any implementational commitments such comparisons might invite. For instance, classicism is clearly not committed to the brain actually having a tape, to there being a controller that is physically distinct from the memory system, to symbols of the sort found in TMs (i.e., two-dimensional arrays of points) and von Neumann machine (high/low voltages), or to any other components characteristic of those architectures. Similarly, in their influential critique of connectionism, Fodor & Pylyshyn (1988) point out that a language of thought is not necessarily serial, is consistent with the 'graceful degradation' observed in connectionist networks, and does not require the explicit coding of rules (pp. 54-60). In short, the more we are told what the language of thought is *not*, the more it resembles the ubiquitous mathematical notion.

3. The Epistemological Utility of Language-Like Representations

On-board representational processes afford some sorts of problem-solving while neglecting others. For example, the default visual category formation mechanisms of human infants are (apparently) good at forming unified categories in the absence of linguistic tags, but cannot form distinct categories without without the aid of such props. This suggests an approach to scaffolding according to which external entities come into play when the limitations of on-board processes need to be surmounted. The proposal explored here, then, is that taking the linguistic structure of on-board representations seriously can provide valuable insight into the limitations of on-board processes, and hence into when scaffolding will arise, and what it is doing when it does.[8] To illustrate I'll consider two examples from the literature on linguistic scaffolding: Numerical reasoning in humans (as discussed by Clark 2008), and the parsing of center-embedded sentences in network architectures (as discussed by Schonbein 2012).

Humans can reason about integer values. How might the mechanisms underwriting this capacity be organized and implemented? Westerners are intimately familiar with place-value notational systems for representing integers: A finite number of instances of numerals drawn from a finite alphabet are arranged in a strict order (the 'places'), each numeral corresponds to an integer (the 'values'), and the quantity represented by the entire string is a function of the value represented by each constituent along with its location in the ordering and the base of the representation. For example, the set of decimal place-value representations is defined by the grammar:

(G_{10}) S → S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

The string ‘129’ (for example) represents the integer 129 as follows:

$$'129' = 9\times10^0 + 2\times10^1 + 1\times10^2 = 9 + 20 + 100 = 129.$$

Obviously, the size of the alphabet (and hence the base) can be changed. For instance, the following grammars define the set of integer representations in octal and binary schemes:

(G_8) S → S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 | 1 | 2 | 3 | 4 | 5 | 6 | 7
(G_2) S → S0 | S1 | 0 | 1

And their respective representations of 129 are:

$$'201' = 1\times8^0 + 0\times8^1 + 2\times8^2 = 1 + 0 + 128 = 129.$$
$$'1000001' = 1\times2^0 + 0\times2^1 + 0\times2^2 + 0\times2^3 + 0\times2^4 + 0\times2^5 + 0\times2^6 + 1\times2^7 = 129.$$

The straightforward way to go about implementing such schemes is to take written language as our cue:

8 The present proposal is not intended to serve as a response to those who argue that the principles of extended cognition are incapable of drawing a satisfactory line between cognitive and non-cognitive processes (e.g., Adams & Aizawa 2009). Instead, it is offered as a version of the uncontroversial claim that understanding human cognitive behavior may require looking at how information is stored, manipulated, and retrieved outside the boundaries of the head.

(i) identify physical states to serve as the realizations of the b primitive symbols, and (ii) assemble an ordered array of elements, each capable of instantiating these states.[9] In a typical digital computer, $b = 2$, the primitive physical states are low and high voltages across a wire, and the array is an ordered set of individual wires. Analogously, in a neural system, we might let '0' be identified with firing rate x of a neuron, and let '1' be identified with firing rate $y \neq x$, and the array is an ordered collection of individual neurons. This array is then used by some further system for its capacity to carry information about integers in a binary place-value format.

Already we see interaction between linguistic structure and implementation. For example, the choice of base (i.e., cardinality of the alphabet) has repercussions for both space (larger alphabets require less space to represent larger values, all things being equal) and reliability (larger alphabets require finer grained discrimination by the users of the physical states realizing tokens of those symbols). Furthermore, given the binary encoding just sketched, there are two ways in which the system will naturally *malfunction*. First, since the array is finite, there will be an *upper bound* that must be dealt with using some other strategy. Second, physical systems are subject to noise, so we have to suppose that for each element in the array there is a *possibility it will misrepresent* the desired value, instantiating a zero rather than a one, or vice versa. Basic probability theory allows us to predict patterns of misrepresentation in such a scheme: Errors of one bit are more likely than errors of two or three bits. So, for instance, if the system is attempting to represent zero using a three-element array, it will misrepresent zero as one, two, or four more frequently than as other values.[10]

These various implementation trade-offs can be viewed as limitations on the representational capacity of the scheme. For example, if a system must use a small alphabet (e.g., binary), and it has limited space, then the upper bound to what it can represent will be lower than a scheme with a larger alphabet. Alternatively, if a system uses a larger alphabet, and the discriminatory capacities of the user of that scheme are fixed, then the error rate will increase due to mistyping. The proposal I'd like to make, then, is that these limitations, arising from the choice of languages and how they are to be realized, are places where linguistic scaffolding is likely to emerge, because it is at these points that the on-board representational scheme encounters difficulty retaining relevant information. In other words, it is when dealing with representing large values and with controlling for likely patterns of error that external entities could be used to *rescue* and *re-present* the potentially corrupted data. For instance, if humans use a binary place-value scheme for representing integers, and this proposal is correct, then appeals to scaffolding should emerge when coping with upper limits on integer contents and with the error patterns described above.

There are, of course, alternatives to the place-value arrays just described. The place-value approach adhered to the principles that (i) every instance of the same symbol type is realized using the same physical pattern (e.g., firing rate x or firing rate y, high voltage or low voltage), and (ii) each instance is realized by the state of a separate component (e.g., neuron, wire). An alternative is to use a single physical state to realize a topology capable of encoding binary strings of arbitrary lengths. For instance, let s be the binary string to be encoded, n the number of digits in that string, and let the digits be numbered beginning with the least significant (i.e., from right to left). Then, we can encode s in variable x using the following method:

9 Obviously 'ordered' does not have to mean 'horizontally concatenated with the least significant digit at the far right'. The order is imposed by whatever components use the physical states for their capacity to carry information about integer quantities.

10 Assuming independent probabilities for each element in the array.

(T_2) Let $x = 0.5$. Then, for $i = 1, 2, 3, \ldots n$, $x(s_i) = \begin{cases} x(s_{i-1}) - \left(\frac{1}{2}\right)^{i+1} \; \textit{if} \; s_i = 0 \\ x(s_{i-1}) + \left(\frac{1}{2}\right)^{i+1} \; \textit{if} \; s_i = 1 \end{cases}$

This method divides the unit line into a hierarchically organized set of points. We start with the midpoint of the unit line (0.5), and then add, at the next lower level, the midpoints of the lines created by dividing at the original segment in two (i.e., 0.25 and 0.75). Next, we add the four midpoints of the next lower set of line segments, created by splitting at 0.25 and 0.75 (0.125, 0.375, 0.625, and 0.875). This process continues *ad infinitum*. The result is a space capable of encoding binary strings of arbitrary length. For instance, beginning at 0.5, the left subspace ($x < 0.5$) encodes '0' while the right ($x > 0.5$) encodes '1'; point 0.75 thus encodes the binary string '1', and represents one. Relative to 0.75, the left subspace ($0.5 < x < 0.75$) encodes an additional '0', and the right subspace ($0.75 < x < 1.0$) encodes an additional '1'; so point 0.625 represents two (i.e., '10'). This process continues to an arbitrary depth, depending on how large the value to be encoded is; 129 is represented by 0.51171875.

It is possible to implement this sort of encoding in a recurrent neural network (with x corresponding to a physical parameter of the neuron, such as 'activation level'). The result is an alternative implementation of G_2 exhibiting different processing characteristics. First, unlike the earlier options, this implementation does not suffer from the hard upper limit imposed by there being a finite number of array elements. Second, the current implementation has a different error profile. Specifically, a user of this encoding must be able to reliably discriminate the complete hierarchy of nested spaces used in any particular string, and the larger the value represented, the more sensitive this capacity must be, i.e., larger strings require higher precision and hence are more susceptible to noise. This means that representations of small values (0, 1, 2, 3, …) should be very reliable, and reliability will gradually decrease as the represented quantity increases; noise has a disproportionate effect on the representation of larger values.

Given the representational limitations of this implementation, scaffolding would be useful in slightly different circumstances than in the case of place-value arrays. Perhaps the most interesting is that, since there is no hard upper limit to the maximum value representable, and noise in the system may be variable (e.g., depending on attention allocated to the task), turning to scaffolding to overcome on-board limitations may be more gradual: Tracking large values precisely requires the use of some sort of scaffolding (on-line or internalized), and tracking small values does not, with a variably messy area in between.

Finally, note that we could pursue similar topological implementations for other bases as well. A particularly interesting case is that of a unary (i.e., single-symbol) grammar:

(G_1) S → S*a* | *a*

The semantics for G_1 is simply that a well-formed string represents the value equal to the number of instances of '*a*' in that string. This grammar could be realized as a place-value scheme (with a significantly higher space cost), but it can also be realized using a simple topological encoding:

(T_1) Let $x = 0$. For each instance of '*a*' in s, $x(s_i) = x(s_{i-1}) + \left(\frac{1}{2}\right)^{i+1}$

As with T_2, this encoding can be straightforwardly implemented, e.g. in a neuron (or ensemble of neurons) with a physical property that varies along the rationals. Furthermore, because the distance between neighboring low values is greater than that between neighboring high values, like binary topological encoding, the device will be most reliable when distinguishing the former, with performance decreasing as quantities increase. The situation is analogous to a variable resistor such as one might find used as a volume control on a stereo.[11] Since the goal is to represent integers, and the resistor varies continuously, let us assume that there are notches (i.e., stable points or attractors) built into the system so that it tends to settle into one of these settings rather than resting in an intermediate location: These are the points defined by T_1. Finally, suppose that (i) the system is subject to noise, and (ii), because of the way this resistor is to be used, it is more important that it be reliable with lower settings than with higher settings. One way to satisfy these design constraints is to place the notches for lower settings farther apart than those for higher values: In this way, there is less chance that ambient noise will perturb the system out of states representing lower values than states representing higher values. This strategy is reflected in T_1 insofar as points representing lower values are farther apart than those representing higher values.

The variable resistor analogy also illustrates just how *different* the topological implementation of T_1 is compared to that of T_2, despite their superficial behavioral similarities. First, if T_2 were implemented in this way, silence would be in the center of the range of the knob's movement. Increasing the volume would then require moving the knob left *or* right, depending on the desired setting. Furthermore, in some cases the knob would have to be moved *through* the silent setting to get to the next increment in volume, e.g., transition from 10 to 11 (i.e., from two to three). In fact, it is easy to see that every time any bit besides the least significant bit is changed from 0 to 1, the system must 'jump' across spatially intervening representations of lower numbers. This suggests that the topological implementation of T_2 is not equally likely to conflate some value pairs differing by one than others, even when both pairs are in the same region of the number line. For example, noise would have a greater chance of causing conflation between 14 and 15 (1110 and 1111) than between 15 and 16 (1111 and 10000).

To sum, I've considered several grammars designed to represent the integers, and sketched some contrasting implementations for those symbol systems (place-value array, topological). We've seen that the choice of grammar (decimal, octal, binary, unary) suggests possible implementations, resulting in different constraints on the behavior of the representational device: Place-value schemes have a strict upper bound on the maximum value representable while topological encodings do not, and each exhibits characteristic patterns of error in the face of noise. Finally, I suggested that it is at these points of failure that we might expect scaffolding to be most useful, since it is at these points where extra-cranial structures could step in to assist in the preservation of relevant information, e.g., through storing that information or through coordinating the on-board scheme in such a way that limitations are attenuated. So, which (if any) of these models best approximates human behavior?

Gallistel et. al. (2006) consider two possible sources of the human ability to represent numbers: By beginning with an underlying discrete representational scheme and approximating continuous values, or through a prior continuous scheme that is constrained so as to represent integers. They

11 The analogy is not perfect, because the present system requires only a discrete representational scheme, while the variable resistor is continuous. For the purposes of this paper I deliberately gloss the distinction between devices that represent integers by quantizing a continuous physical medium (e.g., as digital computers do with continuous voltages) and those that do so by realizing stable states in a continuous representational scheme (e.g., the variable resistor example).

survey various results, including error patterns associated with place-value schemes: Unsurprisingly, humans do not exhibit error patterns corresponding with a binary place-value scheme (p. 252). Instead, humans and other animals exhibit *scalar variability*: the distribution of probabilities of responding when asked to recall a specific value increases as the value to be recalled increases. For example, if to receive food a rat is required to press a lever *n* seconds after a signal, for small values of *n* the rat will attempt to press the lever in a near neighbor hood of *n*, but as *n* increases, the neighborhood around *n* also increases. Gallistel et. al. suggest that scalar variability "seems to be best explained by the assumption that … the reading of a mental magnitude in memory is a noisy process, and the noise is proportional to the magnitude to be read." (p. 251)

Scalar variability is consistent with the topological encodings T_1 and T_2, and so is Gallistel's proposed explanation. Furthermore, there is no evidence that humans exhibit the error patterns predicted by T_2, so, for the purposes of the present paper, we'll treat T_1 – the topological implementation of a unary grammar – as the best candidate model for how humans represent integer quantities.

What does this say about scaffolding? If the proposal that (on-line) scaffolding plays a role in attenuating limitations of on-board schemes (which may themselves be the result of internalization) is correct, then external linguistic aids will help cope with large values. One of Clark's (2008) preferred examples of how linguistic scaffolding offloads on-board representational resources into the environment illustrates just such a situation. Human beings can entertain thoughts about fine-grained relationships between high-valued integers, such as '16748 is one greater than 16747'. This may lead us to presume that the on-board scheme includes either atomic or combinatorial representations for the two values. Appealing to work by Dehaene et. al. (1997, 1999), Clark argues that this is incorrect: Rather, the capacity relies on a trio of other capacities: (i) the ability to distinguish small quantities (e.g., 1, 2, 3, …); (ii) an ability to distinguish distant magnitudes (e.g., 100 being greater than 55); and (iii) an ability to use *numerals*, where it is assumed by the user that different numerals represent distinct quantities. Entertaining the thought that '16748 is one greater than 16747' emerges from the combination of those three capacities: The 'greater-than' relation is captured by ability (ii), that this relation is 'greater by *one*' is handled by ability (i), and extra-cranial linguistic tokens provide external 'pivots' for the behavioral coordination of these abilities.

This form of tripartite coordination fits nicely with on-board representational schemes of the sort illustrated by T_1. Comparing two values requires putting them in order, and if each value is represented in a T_1 type system, then a familiar pattern emerges (i..e, the distance-effect): Reliably ordering two values requires that the difference between those values increase as the values grow in size. This is a straightforward consequence of the fact that the impact of noise is proportional to the values being represented: The system can reliably compare small nearby values (e.g., 2 and 3), but as those values increase, the system needs more distance between them to cope with the effects of noise.

In the example considered by Clark, the goal is to compare two large values that differ by one, so the scheme used by T_1 will have difficulty with this task. Interestingly, this is precisely where scaffolding enters the picture on Clark's (and Dehaene's) account: Language provides an external structure that recasts the unreliable task of finely discriminating between two large numbers into the more reliable task of visually discriminating between two numerals ('7' and '8') and the (small) values they represent. In short, representational schemes such as T_1 implement the abilities to be coordinated (a capacity to distinguish small values and a capacity to distinguish sufficiently distant larger magnitudes), and researchers grant a role to scaffolding precisely where one would expect it to be useful given the limitations of such schemes.

The overall point is not that humans actually represent integer values via T_1-type schemes – that is an issue for cognitive scientists to decide. The point is that by starting with the premise that representational schemes are ubiquitously linguistic, we were able to consider various language-like schemes (decimal, binary, octal, unary) along with their implementations (place-value array, topological), and thereby informally grasp their behavioral characteristics and limitations. Furthermore, it is where these limitations must be confronted that researchers identify a role for on-line linguistic scaffolding. In this way, taking seriously the linguistic structure of mental representation assists in our understanding of when scaffolding enters the picture, and what it is doing when it does.

A second – and much briefer – example can be found in formal language computation by finite-node recurrent neural networks. Automata computing languages of certain complexities require access to a memory wherein symbols are stored and manipulated (Hopcroft & Ullman 1979). Turing machines satisfy this condition by using a tape of unbounded length, and some Turing-equivalent networks work in an analogous fashion, using an unbounded array of nodes to implement a tape (Franklin & Garzon 1991). However, neither strategy is available to finite-node networks; instead, these networks pursue a topological approach similar to those described in the previous example: Constituent structure is encoded in self-similar phase spaces, such as Cantor sets or Sierpinski triangles. In at least some networks (e.g., Tabor 2000), this method of encoding means that parsing center-embedded sentences – in contrast to other sorts of recursive structures – requires descending into regions of space that demand finer-grained discrimination on the part of the user of the representations (Schonbein 2012).

As in the case of T_1 and T_2, above, these different implementation strategies result in different patterns of behavior when subjected to noise. Suppose, for instance, that noise is applied to the to the read/write head of a Turing machine so that there is a chance the controller misinterprets the current symbol. In such a situation, each symbol is impacted independently of the others; consequently, the *internal structure* of the string is not relevant to misrepresentation: The first symbol in the string is just as likely to be misread as the fiftieth, even if some of those symbols are part of center-embedded clauses. In contrast, in the Cantor set encoding, noise disproportionately effects symbols appearing in center-embedded clauses, because working with those symbols requires finer grained discriminatory capacities on the part of the system using those representations. In other words, the tracking of embedded strings requires additional digits of precision, and noise has a greater chance of disrupting higher-precision digits than lower-precision digits. In short, purely by virtue of the way the same linguistic structure is encoded, the two systems – TMs and a network using a Cantor set – exhibit different patterns of misrepresentation.

All else being equal, if humans use TM-style representations, errors in parsing sentences should occur at the same rates for center-embedded sentences as for left- and right-branching sentences, but if humans use network-style representations, they should have selective difficulties with center-embedded in contrast to left- and right-branching sentences. Furthermore, these differences suggest different roles for linguistic scaffolding: If humans use network-style representations, and scaffolding emerges when on-board codes have difficulty retaining relevant information, then they will be likely to resort to the use of external linguistic entities (e.g., written or spoken words) when faced with tasks that require the parsing of center-embedded strings.

Unlike the case of numerical comparisons (where we could follow through with information about how researchers have brought scaffolding into the picture), to my knowledge comparable work has not been done on scaffolding and the parsing of center-embedded sentences. However, in this case we can informally run an intuition pump by asking the reader to appreciate the potential utility of pencil and paper when parsing the following sentence:

The dog the girl the boy the driver avoided kissed fed barked.

Assuming there is agreement that external scaffolding is useful in this case (e.g., by allowing for the rearrangement of the constituents of the sentence), the parsing of center-embedded sentences provides another example of how insights into scaffolding are facilitated by taking seriously the linguistic structure of on-board representational schemes along with their possible implementations.

4. Conclusion

In this paper I've argued that language-like representations are not pernicious. Instead, not only may they be unavoidable (section 2), they are an important tool for understanding the 'whens' and 'whys' of linguistic scaffolding (section 3).

We can gain a general perspective on this conclusion by drawing on a distinction between explicit and implicit representation, as articulated by Kirsh (1990) and Clark (1992). A representation is explicit if it makes the information it contains readily available to a consumer of those representations; in contrast, a representation is implicit if the information it contains is not so available. For instance, both a standard well-formed formula of predicate calculus and a Gödel number may encode the same logical structure, but, relative to the human visual system, only the former explicitly represents that information. Similarly, in a Turing machine, the constituent structure of a string is explicitly represented relative to the human visual system, while the same information is only implicitly represented by states in a finite-node recurrent network.

Assuming that we keep the consumer fixed, using information encoded implicitly requires that the representation be *manipulated* so that the required information becomes explicit, and hence readily available. For instance, to access the logical structure encoded in a Gödel number, we may use pencil and paper to recode that information into standard wff form. Similarly, the bulk of the work involved in demonstrating the Turing-equivalence of automata (e.g., finite-node recurrent networks) has been in making explicit the information contained in representations so that inter-automata mappings could be constructed.

Explicitness may also be a matter of degree. For instance, with sufficient practice, humans may be able to work with some Gödel numbers directly, while others still require recoding. Similarly, a variable-resistor-style representational scheme (T_1 or T_2) for encoding integers has the *theoretical* capacity to represent values of arbitrary magnitude, yet, as noted above, noise results in restrictions on the reliability of representations of higher values. As the value to be represented increases, the availability of that information decreases. A parallel story holds for center-embedded clauses in recurrent network representations. Because whether a representation is explicit may vary according to the information it carries, the need to recode such representations – e.g., translating a Gödel number into a traditional wff, a network state into a TM string, a variable-resistor-style integer representation into a decimal one, or a center-embedded sentence into one that branches left or right – may also vary across contexts. Recoding occurs when the explicitness of on-board representations breaks down, requiring that the information be rendered explicit using other means.[12]

This provides a general perspective on linguistic scaffolding and its relation to on-board systems of representation: When an on-board representational scheme can no longer effectively make explicit (relative to internal consumers) information necessary to performing some task, this information is *rescued* through an explicit (relative to the same or to other consumers) external

12 That 16748 is one greater than 16747, for instance.

recoding. The information contained in external recodings then drives behavior by coordinating on-board resources, which may include the very same representational mechanisms that led to the use of scaffolding in the first place. In this way, Kirsh's and Clark's analysis of the implicit/explicit distinction allows for a fruitful broad conceptualization of the main theme of this paper.

References

Aydede, M. (1997). 'Language of thought: the connectionist contribution.' *Minds & Machines* 7: 57-101.

Bechtel, W. and A. Abrahamsen (2002). *Connectionism and the Mind*. Oxford, Blackwell Publishers, Ltd.

Berlekamp, E. R., Conway, J. H., & Guy, R. K. (1982). *Winning Ways for your Mathematical Plays*. London, Academic Press.

Bournez, O. and M. L. Campagnolo (2008). 'A survey on continuous time computations', in: *New Computational Paradigms: Changing Conceptions of What is Computable*. S. B. Cooper, B. Lowe and A. Sorbi. (Eds.) New York, Springer-Verlag: 383-423.

Chomsky, N. (1963). 'Formal Properties of Grammars'. *Handbook of Mathematical Psychology*. R. D. Luce, R. R. Bush and E. Galanter. (Eds.) 2: 323-418.

Churchill, A. (n.d.). '*Magic: The Gathering* is Turing complete.' Retrieved 13 September 2012, from http://www.toothycat.net/~hologram/Turing/index.html.

Ciesinger, J. (1988). 'Neural nets as powerful as Game of Life (as powerful as Turing Machines).' SIGACT News 19(3-4): 46-47.

Clark, A. (1992). 'The presence of a symbol.' *Connection Science* 4(3 & 4): 193-205.

Clark, A. (1993). *Associative Engines*. Cambridge, MA, MIT Press.

Clark, A. (1997). *Being There: Putting Brain, Body and World Together Again*. Cambridge, MA, MIT Press.

Clark, A. (1998). 'Magic words: how language augments human cognition.' *Language and Thought: Interdisciplinary Themes*. P. Carruthers & J. Boucher (Eds.). Cambridge, Cambridge University Press: 162-183.

Clark, A. (2004). 'Is language special? Some remarks on control, coding, and co-ordination.' *Language Sciences* 26: 717-726.

Clark, A. (2008). *Supersizing the Mind: Embodiment, Action, and Cognitive Extension*. Oxford, Oxford University Press.

Dehaene, S. (1997). *The number sense: how the mind creates mathematics*. Oxford, UK, Oxford University Press.

Dehaene, S., Spelke, E., Pinel, P., Stanescu, R., Tsivkin, S., (1999). 'Sources of mathematical thinking: behavioral and brain-imaging evidence.' *Science* 284: 970-974.

Dennett, D. C. (1991). *Consciousness Explained*. Boston, Little, Brown & Co.

Fodor, J. A. (1975). *The Language of Thought*. Cambridge, MA, Harvard University Press.

Fodor, J. A. (1987). *Psychosemantics*. Cambridge, The MIT Press.

Fodor, J. A. and Z. W. Pylyshyn (1988). 'Connectionism and cognitive architecture: A critical analysis.' *Cognition* 28: 3-72.

Franklin, S. and M. Garzon (1991). 'Neural Computability'. in *Progress in Neural Networks*. O. M. Omidvar (Ed.). Norwood, NJ, Ablex. 1: 127-146.

Gallistel, C. R., Gelman, R., & Cordes, S. (2006). 'The cultural and evolutionary history of the real numbers'. in *Culture and Evolution*. Levinson, S. & Jaisson, P. (Eds.) Cambridge, MA, MIT Press: 247-273.

Garnier, S., Combe, M., Jost, C., & Theraulaz, G. (2013). 'Do ants need to estimate the geometrical properties of trail bifurcations to find an efficient route? A swarm robotics test bed.' *PLoS Computational Biology* **9**(3): 1-12.

Grunbaum, B. & G. C. Shephard (1987). *Tilings and Patterns*. New York, W. H. Freeman and Company.

Hopcroft, J. E. & J. D. Ullman (1979). *Introduction to automata theory, languages, and computation.* Reading, MA, Addison-Wesley.

Hutchins, E. (1995). *Cognition in the Wild*. Cambridge, MA, MIT Press.

Kirsh, D. (1990). 'When is information explicitly represented?' *Information, Language, and Cognition, Vancouver Studies in Cognitive Science, Vol. 1*. P. Hanson. (Ed.) Oxford, Oxford University Press: 340-365.

Moore, C. & S. Mertens (2011). *The Nature of Computation*. Oxford, UK, Oxford University Press.

Plunkett, K., Hu, J.-F., Cohen, L. B. (2008). 'Labels can override perceptual categories in early infancy.' *Cognition* 106: 665-681.

Rowlands, M. (1999). *The Body in Mind: Understanding Cognitive Processes*. Cambridge, Cambridge University Press.

Rowlands, M. (2009). 'Situated representation'. *The Cambridge handbook of situated cognition*. P. Robbins & M. Aydede (Eds.). Cambridge, UK, Cambridge University Press: 117-133.

Rumelhart, D. E., Smolensky, P., McClelland, J.L., & Hinton, G.E. (1986). 'Schemata and sequential thought processes in PDP models'. *Parallel Distribute Processing: Explorations in the Microstructure* *of Cognition, Volume 2: Psychological and Biological Models*. J. L. McClelland, D. E. Rumelhart (Eds.). Cambridge, MA, MIT Press: 7-57.

Scarle, S. (2009). 'Implications of the Turing completeness of reaction-diffusion models, informed by GPGPU simulations on an XBox 360: Cardiac arrhythmias, re-entry and the Halting problem.' *Computational Biology and Chemistry* 33: 253-260.

Schonbein, W. (2012). 'The linguistic subversion of mental representation.' *Minds & Machines* 22: 235-262.

Siegelmann, H. T. and E. D. Sontag (1994). 'Analog computation via neural networks.' *Theoretical Computer Science* 131: 331-360.

Tabor, W. (2000). 'Fractal encoding of context-free grammars in connectionist networks.' *Expert Systems* 17(1): 41-56.

Wheeler, M. (2004). 'Is language the ultimate artefact?' *Language Sciences* 26: 693-715.

Wheeler, M. (2007). 'Continuity in question: An afterword to Is Language the Ultimate Artefact?' *The Mind, the Body and the World: Psychology after Cognitivism?* B. Wallace, A. Ross, J. Davies & T. Anderson (Eds.). Exeter, UK, Imprint Academic: 298-305.

Wheeler, M. & Clark, A. (1999). 'Genic representation: Reconciling content and causal complexity.' *British Journal for the Philosophy of Science*. 50: 103-135.