Minds & Machines (2012) 22:235–262
DOI 10.1007/s11023-012-9275-1

The Linguistic Subversion of Mental Representation

Whit Schonbein

Received: 18 May 2010 / Accepted: 15 May 2012 / Published online: 2 June 2012

Abstract Embedded and embodied approaches to cognition urge that (1) complicated internal representations may be avoided by letting features of the environment drive behavior, and (2) environmental structures can play an enabling role in cognition, allowing prior cognitive processes to solve novel tasks. Such approaches are thus in a natural position to oppose the 'thesis of linguistic structuring': The claim that the ability to use language results in a wholesale recapitulation of linguistic structure in onboard mental representation. Prominent examples of researchers adopting this critical stance include Andy Clark, Michael Wheeler, and Mark Rowlands. But is such opposition warranted? Since each of these authors advocate accounts of mental representation that are broadly connectionist, I survey research on formal language computation in artificial neural networks, and argue that results indicate a strong form of the linguistic structuring thesis is true: Internal representational systems recapitulate significant linguistic structure, even on a connectionist account of mental representation. I conclude by sketching how my conclusion can nonetheless be viewed as consistent with and complimentary to an embedded/embodied account of the role of linguistic structure in cognition.

Keywords Situated cognition · Language · Artificial Neural Networks · Connectionism · Mental representation

Introduction

Many philosophers hold that linguistic competence has significant consequences for the structure of mental representation. Prominent examples include Dennett's theory of consciousness (Dennett 1991) and Fodor's language of thought hypothesis (Fodor

W. Schonbein (✉)
Department of Philosophy, College of Charleston, 14 Glebe Street, Charleston, SC 29424, USA
e-mail: 71whit@gmail.com

1975). Despite substantive differences in both the justification for positing mental representations and the role those representations are supposed to play in cognition, such views share the position that representational systems central to our mental lives recapitulate structural features of language. Call this the 'thesis of linguistic structuring' (TLS).[1]

Approaches to language based on embedded and embodied cognition (EEC) are naturally poised to resist the TLS. For example, traditional cognitive science assumes that our brain maintains a comprehensive representation of the external world. In contrast, a central EEC principle is that cognitive systems often eschew complicated internal models, and instead sparsely represent features of the environment, consulting the world only when further information is needed (Brooks 1991; Clark 2008, p. 15).[2] In this way the complexity of the internal representational system is reduced.

This principle puts pressure on the TLS. Complicated internal representational systems are expensive in terms of installation, operation, and maintenance. Granting that public language has certain complex structural properties (e.g., constituent structure), is it really necessary for an individual to internalize these properties, mirroring them in a system of mental representation? Perhaps it would be more efficient to leave those features in the world, relying on a less expensive on-board system of representation, and consulting these structures only when needed.

This suggestion is rendered additionally attractive when conjoined with another principle of EEC, that appropriately structured features of the environment can transform existing forms of cognition or even enable new ones. Long division provides an anecdotal example: A child cannot do long division in her head, but she can write numerals on a page in a prescribed way, follow rules for manipulating those symbols on that page, and thereby perform long division. External language may thus extend prior forms of cognition in new and significant ways. And if this is the case, it may be the very 'externality' of the properties that makes all the difference, as those properties form stable, easily (re)identified markers around which to coordinate individual and group behavior. Mirroring structural features of language seems almost counterproductive from this perspective.

In short, the EEC approach to language invites opposition to the TLS. It is therefore no surprise that proponents of EEC view the TLS with skepticism. For instance, while granting that internal representations recapitulate grammatical categories (e.g., nouns vs. verbs), Michael Wheeler (2007) withholds the stronger claim that they exhibit combinatorial structure. Similarly, drawing on early work in the second wave of connectionist modeling, Mark Rowlands (1999, 2009) argues that we should resist the urge to infer from the fact that written and spoken language exhibits logical structure to the conclusion that mental representations also have this structure. And in a series of papers and books extending back a decade, Andy Clark proposes that language does not restructure internal representations; rather, language is "an external resource that complements—but does not profoundly alter—the brain's own basic modes of representation and computation." (1998,

[1] The TLS is elaborated, below.

[2] This principle is often summarized as, 'the world is its own best model'.

p. 167). On this view, linguistic structure modifies cognitive processes, but only as an external feature of the environment.

There are two distinct issues at play here. The first concerns the *role* that language plays in cognition: To what degree, if any, does language play a role in thought? The second concerns the *format* of the on-board representational system: To what degree, if any, does it share properties with linguistic representations? The answers to these questions are independent of each other. On the one hand, someone could adopt the position that language is an essential component of many cognitive processes while simultaneously asserting that mental representations associated with language use are unstructured—linguistic structure functions purely as a form of *external* scaffolding. On the other hand, someone could hold that linguistic structures play a very limited role in cognition, despite being recapitulated in all of their fine-grained detail in a complex system of mental representation. The TLS is solely concerned with the issue of format: How much, if any, of linguistic structure is internally recapitulated, and how much can be left in the extracranial environment?

Answers to this question fall along a continuum. At one extreme is the claim that there are no mental representations related to language (perhaps because there are no mental representations at all), and hence there is no recapitulation of structure. A less radical position is to allow that there are mental representations dedicated to language use while denying they recapitulate any significant linguistic structure. This is Clark's position, as I interpret it. One might go a bit further, as Wheeler (2007) does, and allow that the system of language-related mental representation recapitulates the general grammatical categories of external language (noun, verb, etc.) without exhibiting constituent structure. These positions are both treated here as in opposition to the version of the TLS I hope to defend, namely, that mental representation does recapitulate the constituent structure of language.

There are a number of strategies for mounting this defense. For example, Gary Marcus' work on connectionist networks and grammatical rules suggests that some form of linguistic restructuring of internal states is necessary (Marcus et al. 1999; Marcus 1999). Alternatively, while the debate is of course not settled, the literature on linguistic nativism contains plenty of ammunition one might deploy in support of a version of the TLS. However, in this paper I hope to present a novel argument, one that builds directly on the assumption, shared by the various parties, that 'the brain's own basic modes of representation and computation' are connectionist. From this starting point, I argue that research on formal language computation in artificial neural networks shows that language use requires that both internal representations and operations recapitulate the structure of language.[3]

The structure of the paper is as follows. Since Clark's version of the EEC case against the TLS has recently been updated (Clark 2008), I use it as a starting point, providing an overview of his argument in "The EEC Case Against Linguistic Structuring". In "Language and Biologically Basic Representation" I articulate and defend my reply to Clark's argument. In "Extending the Argument", I show how my argument can be extended to address Rowlands' and Wheeler's versions of the

[3] This thesis is intended to be independent of any claims concerning language nativism.

anti-TLS position. Finally, I conclude by providing several illustrations of how these results are nonetheless complementary to the EEC approach to understanding cognition.

The EEC Case Against Linguistic Structuring

According to the TLS, a significant portion of mental representation reiterates certain structural properties of public language. In the version discussed here, it is the constituent structure of language that is repeated in a system of mental representation (Wheeler 2004, p. 710; Clark 2004, p. 722). Clark's embodied alternative holds that these structural properties need not be recapitulated in the brain's own representational system. Instead, he argues that brains "can use [language] without radically altering their basic modes of representation and computation." (Clark 2004, p. 720). That is, his view

> depicts language as an external artefact designed to complement, rather than recapitulate or transfigure, the basic processing profile we share with other animals. It does not depict experience with language as a source of profound inner re-programming … (Clark 1998, p. 169)

Or, as he puts it in his (2008),

> perhaps the brain represents these potent real-world items [i.e., words and sentences] in much the same way it represents anything else. In that case, language need not reorganize neural coding routines in any way that is deeper or more profound than might occur, say, when we first learn to swim or to play volleyball. (Clark 2008, p. 56)

Or, as he puts it in his (2004),

> a representation of structure is not thereby … a structured representation. Just 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)[4]

The proposal is clear: The brain need not utilize a representational scheme with constituent structure because it can use its own 'biologically basic' modes of representation to do the job, leaving the structure in the world.

What are these 'biologically basic' forms of representation? On Clark's view, they are broadly connectionist (with an emphasis on a dynamical systems theory

[4] The title of this paper is jointly inspired by Clark and Rowlands. Clark views external language as something that "does not radically transform the inner processing economy, so much as subvert it to new ends," (2004, p. 720), while Rowlands' stated goal is to "subvert a particular pre-theoretical picture of the mind." (1999, p. 149).

(DST) perspective).[5] (Clark 1993) The basics of such views are well known, but here is an extremely brief rehearsal:[6] On a basic connectionist account, networks are collections of nodes joined by weighted connections, and the nodes can take on different levels of activation. Occurrent internal representations are vectors of activation distributed across nodes. Computational operations can be interpreted as roughly associative, involving processes of pattern-completion and pattern-categorization. From a DST perspective, (occurrent) representations are points in a high-dimensional state space, where the topological structure of this space is determined by the weight configuration. Operations on these representations result in trajectories through state space over time. The connectionist/DST account is typically taken to stand in opposition to the well-known language of thought theory: Thoughts are composed of constituent atomic concepts ('words') built up according to rules of combination ('grammar'), and reasoning is understood as operations on these atomic and molecular language-like representations. Thus, Clark's basic modes of representation fit nicely with his resistance to the TLS—they are not language-like, and language use does not require that they be restructured to become language-like.

The support for Clark's anti-TLS proposal is empirical, and the research he surveys motivates both of the claims introduced above. First, that language qua external entity expands the problem-solving repertoire of our basic cognitive capacities; and second, that this explanation does not require the recapitulation of linguistic structure in internal representation. For purposes of brevity, I here summarize only two of the studies Clark surveys, indicating how they relate to these two claims:

1. Chimpanzees and same-different relationships (Thompson, Oden, & Boysen 1997). Chimpanzees can categorize pairs of objects according to whether both members are of the same type (e.g., shoe–shoe) or of different types (e.g., shoe-cup). But despite being able to identify the properties of sameness and difference, they have difficulty with a 'conceptual matching task': Given a pair of objects of the same type (/different) type, which of two additional pairs of objects also have that property of sameness (/difference)? For example, when shown the pair 'cup-cup', and then given a choice between the pairs 'shoe-shoe' and 'cup-bottle', the animal is supposed to pick the 'shoe-shoe' pair, as it shares the property of *sameness* with the original pair. Researchers found that only those chimpanzees trained to use plastic tags (e.g., a blue diamond and a green circle) to mark sameness and difference during the original discrimination task were able to solve the conceptual matching task.

Clark suggests that those chimpanzees that used the tags in the original task were able to mentally recall those tags when given the conceptual matching problem, 'virtually' marking all three pairs with internal, surrogate tags. This has the effect of reducing the problem to be solved to the original task: The animal simply has to

[5] In this summary I focus only on the internal (brain-based) representational scheme, since the issue at hand is whether this format is changed significantly by language. Clark extends this account to handle embodiment and embedding, and this is one of the primary reasons for adopting a DST framework (e.g., Clark 2008, pp. 24–28; Spivey and Richardson 2009, pp. 384–385).

[6] See Bechtel and Abrahamsen (2002) for more information.

look for the pair that has the same tag as the target pair. This illustrates the two central themes of Clark's proposal. First, the chimpanzee's brain has the onboard resources to solve the same-different task, but cannot solve the conceptual-matching task. However, by supplementing the environment with linguistic items, these already-present capacities can be brought to bear on the latter task—the role of the labels is to redirect onboard resources in such a way that the basic modes of representation and computation can parse the world in new ways (Clark 2008, p. 49). Second, there is no reason to suppose that this requires that the basic structure of those onboard resources is changed in any significant way—it is still a pattern-matching process.[7]

2. Arithmetical thought (Dehaene 1997; Dehaene et al. 1999). A further source of evidence comes from Dehaene's theory of arithmetical thought. A traditional way of explaining how we entertain the thought that 98 is one greater than 97 (to use Clark's example) is to posit a mental sentence with three syntactically arranged constituents—the concepts of 98, 97, and GREATER-THAN. Drawing on brain lesion, neuroimaging, and behavioral studies, Dehaene argues that there are in fact *no* such constituents. Instead, the ability to entertain the thought emerges from the interaction of three capacities: (1) A capacity to distinguish small specific quantities (1,2,3,…), and to recognize basic ordinal relations between them (e.g., more-than); (2) a capacity to distinguish magnitudes in a non-specific way (e.g., being able to judge that a group of 20 objects is less than a group of 50, without having any specific quantities in mind); and (3) a (learned) capacity to use number words and numerals in a language, with the assumption that each names a distinct quantity (although that specific quantity need not be represented in the mind).

These three capacities join together to allow one to entertain thoughts about specific large quantities, as in the case of holding that 98 is one greater than 97. Our biologically basic capacity (2) allows us to represent the fact that one quantity is greater than another; the basic capacity (1) allows us to represent the fact that one quantity is greater than another *by one*; and the presence of a system of number words in a public language with which we are competent—capacity (3)—gives us the ability to harness the first two representational components into the claim that whatever quantity is designated by the numerals '98' or number-word 'ninety-eight' is exactly one more than that quantity designated by '97' or 'ninety-seven'.

On Dehaene's account, entertaining thoughts about large quantities does not require internal representations containing, as components, concepts of the specific quantities:

> What matters for present purposes is that there may be no need to posit (for the average agent), in addition to this coordinated medley, any further content-matching internal representation of, say, 98-ness. Instead, the presence of actual number words in a public code (and of internal representations *of those*

[7] Of course, one might object that the linguistic items have *no* linguistic structure, so there is no such structure for the onboard system to recapitulate. A proponent of the TLS can thus agree with Clark in this instance, insisting that other tasks will require the replication of structure. In "Language and Biologically Basic Representation" I consider a task that arguably does require the replication of linguistic structure.

> *very public items*) is itself part of the coordinated representational medley that constitutes many kinds of arithmetical knowing. (Clark 2008, p. 52)

Rather than representations with specific constituents, we have a "hybrid" representation that "includes, as a co-opted proper part, a token … of a conventional pubic language encoding ("ninety-eight") appropriately linked to various other resources (e.g., some rough position on an analog number line)." (Clark 2008, p. 53) As in the previous cases, language plays a crucial role in enabling biologically basic systems to achieve a goal (in this case the representation of specific large quantities). Furthermore, the internal representational mechanisms clearly do not recapitulate the structure of external language, because there are no such internal representations.

Clark's argument against the TLS is thus two-pronged. First, the research on numerical reasoning suggests that at least in some cases there are no explicit mental representations corresponding external linguistic items. That is, the mental lexicon is smaller than one might think. The second prong—which is the focus of this paper—is to grant that there are mental representations dedicated to language use, but deny that these have constituent structure. This strategy is illustrated by both the conceptual matching task and the numerical reasoning research: Each suggests that the causal role of linguistic structure with respect to our cognitive capacities is better conceived of as residing in the extracranial environment, since it is the externality of language that is doing such important work.

Language and Biologically Basic Representation

Is Clark correct in asserting that facility with linguistically structured environmental entities has no significant consequences for the structure of internal representations underwriting this capacity? Given his endorsement of connectionist representations, I investigate the consequences for a connectionist representational system when that system must represent the structure of language. Now, one potential criticism of Clark's argument is that many of the studies he cites either do not rely on linguistic structure, or are otherwise consistent with the TLS.[8] So to be safe we need a task that *requires* a system make use of linguistic structure; that way we can assess

[8] I do not pursue this criticism here, but such an argument might make the following claims: The conceptual matching task study described in the main text uses colored plastic tags, which are symbols whose internal structure is irrelevant to the task. Similarly, the studies reported in Boysen et al. (1996) and Plunkett et al. (2008) do not require the tracking of internal structure. In contrast, the study reported in (Hermer-Vazquez et al. 1999) indicates that language interacts with spatial memory, but is consistent with internal representations becoming linguistically structured through language acquisition. This leaves the work on arithmetical thought carrying the weight of Clark's rejection of the TLS, and that work does not show that internal representations are not linguistically structured so much as that the set of mental representations has fewer members than previously thought. This case-by-case critique of Clark's argument would require significant elaboration, but the methodological point—that assessing Clark's claim requires looking at a task that requires the tracking of linguistic structure—remains.

whether any significant recapitulation appears in the internal representational system. A good candidate, I propose, is the recognition[9] of members of a formal language.

There are several reasons for considering this task. First, it is arguably a necessary condition on linguistic competence that any competent individual be able to distinguish between grammatical and ungrammatical sentences (in most cases). Second, recognizing a language requires no special training beyond being competent in the language, so it cannot be claimed to be a special case of linguistic expertise. Third, the capacity is necessarily grounded in internal representations: Since not all external linguistic tokens are grammatical, it cannot be off-loaded onto environmental scaffolding. Fourth, the task requires the representation of constituent structure insofar as it requires subjects to keep track of intra-sentence dependencies between linguistic types across intervening clauses. Fifth, the task can be considered purely formally, in isolation from questions about semantics, allowing us to focus purely on linguistic structure. Sixth, there is a substantive literature on how Clark's biologically basic modes of representation—i.e., artificial neural networks—recognize formal languages. This means we can retain, as a working assumption, the view that the basic modes of representation and computation are connectionist, and then ask whether successful task performance requires any noteworthy restructuring of that system.

Languages can be arranged according to their structural complexity (Chomsky 1963). There is some dispute over the complexity of natural language, but it is generally agreed to be at least context-free (Pullum and Gazdar 1982; Shieber 1985). So our question becomes: What sort of organization must a connectionist network assume in order to recognize at least a context-free language? I'll try to answer this question in three stages. First, I'll look at results concerning the organization of state space in networks that recognize the least-complex type of language—regular languages. Second, I'll summarize Tabor's (2000) example of a network that computes a context-free language. Finally, I'll briefly look at how these results may extend to other types of language. The conclusion I wish to motivate is that (1) the state spaces of networks will replicate the constituent structure of the languages they recognize, and (2) the networks adopt a common strategy to achieve this structuring, suggesting that this solution generalizes to other types of network and to other languages. To be clear, I am not claiming that the results summarized here constitute the *only* way for networks to recognize context-free languages; rather, the claim is that taken together, these results provide evidence that networks will recognize such languages in this way, and that the burden of proof is thus on detractors to articulate and defend alternatives that do not recapitulate structure.

[9] To say that a system recognizes a language is just to say it can distinguish between strings that are members of the language and those that are not, i.e., distinguish between grammatical and ungrammatical sentences.

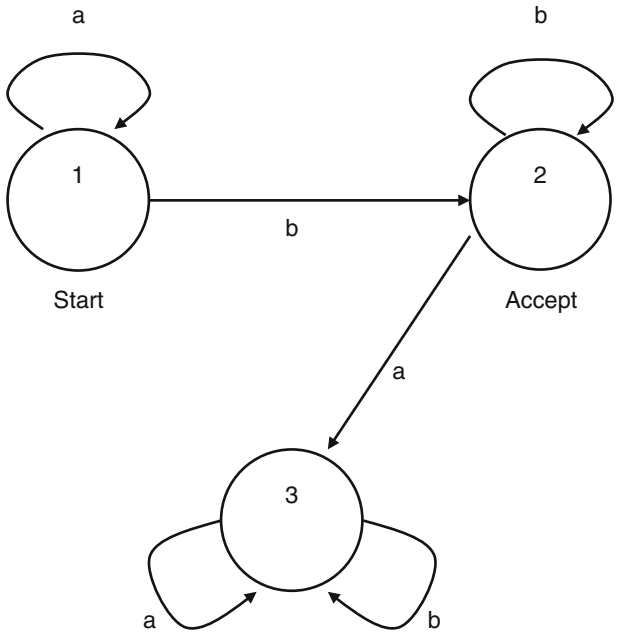

Fig. 1 A DFA that computes the language $a^n b^m$, where $n \geq 0$ and $m > 0$

Regular Languages

A regular language is one that can be recognized by a deterministic finite automaton (DFA). Taking the alphabet to be {*a*, *b*}, some examples include (1) the set of all strings with an even number of *a*s and (2) the set of strings consisting of zero or more *a*s followed by one or more *b*s. DFA can be represented as collections of nodes with conditional transitions between them. One node is designated as the starting node and another is designated as the acceptance node. The automaton is presented with a candidate string, one symbol at a time, and the active node changes according to the conditional transitions connecting nodes. If the system is in the acceptance state when the end of the string is reached, the input is determined to be a member of the language; otherwise, it is rejected as ungrammatical.

Figure 1 depicts a sample DFA that recognizes the second example given above. For example when given the strings *aaabbb*, *bb*, or *aaaaaab*, the automaton will accept the string. When given the strings *abab*, *aaaa*, or *aaabbba*, the system stops in either state 1 or 3, and the string is rejected.

For each regular language there exists a DFA with a minimal number of nodes to recognize the language—a DFA with fewer nodes cannot recognize the language, and a DFA with more has redundant nodes. Omlin and Giles (1996) and Casey (1996) demonstrate that, if a network with a finite number of nodes recognizes a regular language, then it must organize its state space in accordance with the minimal DFA for that language. In other words, if you are given a network that recognizes the language and analyze the state space, you will find that it is divided

into regions corresponding to the states of the minimal DFA, and transitions between those regions correspond to the transitions between states in that DFA. Furthermore, no networks that recognize the language lack a state space so organized.[10]

This result suggests a lower limit for state space organization in the sense that processing even simple languages requires that the internal structure of a network be correspond with formal aspects of the language being computed. This in turn suggests that computing more complex languages will require more complex—yet nonetheless principled—structuring of state space.

Context-Free Languages and Dynamical Automata

DFA have no way of storing information about symbols previously encountered except by adding further nodes. This restriction means they cannot track dependencies between components of an input string. For example, consider the language consisting of all strings beginning with one or more *a*s and followed by an equal number of *b*s: (*ab*, *aabb*, *aaabbb*, …}. A small amount of time with a pencil and paper should be enough to convince us that a DFA cannot recognize this language. The problem is that the only way to store the information that *n a*s (or *b*s) have been encountered is to use *n* distinct nodes, so in the general case where the number of symbols is not known in advance, an infinite number of nodes is required to handle every possible input. This is clearly unacceptable if we want our model to be a model of a cognitive process.

The traditional way to address this problem is to expand the DFA by adding a memory store in the form of a tape divided into squares, where each square can contain one symbol. The DFA is given a read/write head that can be moved backwards and forwards in one-square increments, reading and writing symbols from and to the tape. Rather than react only to the current input symbol and current state, the system also reacts to the current symbol being read from the tape; similarly, rather than simply transitioning to another state, the system can also write a symbol to the tape and move the head forward or backwards.

The tape is usually of unbounded length, which means that if the system needs more memory, there is always another blank square available. However, different types of automata have different sorts of restrictions on how the controller accesses memory. For example, in pushdown automata (PDA) the controller is limited to accessing items in memory, like a stack of cafeteria trays, in reverse of the order they were stored. To store an item in memory, a symbol is 'pushed' onto the top of the stack, and to get at a symbol stored in memory, symbols are 'popped' off the top of the stack (and discarded). PDA compute the class of context-free languages. The question is thus how to augment a connectionist network so that it has the functional equivalent of a stack to make use of in processing syntactic information.

[10] The 'must' here *is* logical necessity (Casey 1996). However, given that the authors demonstrate this result only for a specific type of network architecture (second order recurrent neural networks) and complexity class of languages (regular languages), the present emphasis is that these results provide reason to suppose that the computation of more complex languages by similar architectures will also involve linguistically structured state spaces.

We cannot solve the problem by adding a tape and read/write head to a network, because that violates our criterion that only 'biologically basic' representational forms be used. It also will not suffice to construct a network that has a subnet that implements a tape and read/write head, as that solution requires an unbounded number of nodes: As more memory is required, additional nodes are added to the network to create the functional equivalent of more squares and methods to access them. We want a network with a finite number of nodes. The DFA networks point toward a solution: Structure state space so that it can function as a memory store.[11]

Here's a very simple example of such a structured space (adapted from Moore 1998, p. 100). The alphabet is $\{a, b\}$, and the language is the set of strings where the number of *a*s is equal to the number of *b*s. The system starts at $x = 0.5$, and processes input symbols according to two rules. If the current input is *a*, then the location on the unit line is $f(x) = x * 0.5$; and if the current input is *b*, then the location on the unit line is $f(x) = x * 2$. If, at the end of the input string, $x = 0.5$, then the string is recognized; otherwise it is rejected.

This system can be realized in a neural network (Fig. 2).[12] Furthermore, this system—like the DFA networks—computes by using a well-structured state space. The rules governing the behavior of the network divide hidden unit space into a series of points (0.5, 0.25, 0.125, …). These points are related to one another in such a way that transitioning between any two neighboring points uses precisely the same rule. This allows the network to use that structure to store information about the ratio of *as* to *bs* in the input string.

The simplicity of this type of state space places limits on the computational power of systems that use it. However, if the state space is structured in the form of Cantor set, the resulting class of automata can compute the set of context-free languages (Moore 1998). So we know that there is no context-free language that finite-node networks cannot recognize.

Tabor (2000) extends Moore's results into multiple dimensions, and illustrates how the resulting 'dynamical automata' can be implemented in connectionist networks. Tabor's central example involves a two-dimensional state space structured as a Sierpinski triangle. The Sierpinski triangle is formed by taking a right-angled triangle and removing the middle triangle, leaving three sub-triangles. The middle triangle is then removed from each of these, and so on (Fig. 3). Tabor demonstrates how a network with a state space structured in this way can recognize a context-free language with center embedding (Table 1).

S, *A*, *B*, *C*, and *D* are non-terminal symbols; terminal symbols are *a*, *b*, *c*, and *d*. The symbol *e* is empty. The grammar has two 'basic' strings: *abcd* and *abad*. It then allows for (1) further iterations of these strings to be appended, and (2) any occurrence of the basic strings to be center-embedded. So, for example, the following strings are all members of the language (the parentheses are for illustrating the center-embedded strings, and are not part of the language).

[11] In addition to the research described in the main text, other relevant studies see also (Christiansen and Chater 1999; Bodén et al. 1999; Rodriguez et al. 1999; Siegelmann 1999).

[12] This implementation is based on the architecture used in Tabor's (2000) similar construction for a two-dimensional state space.

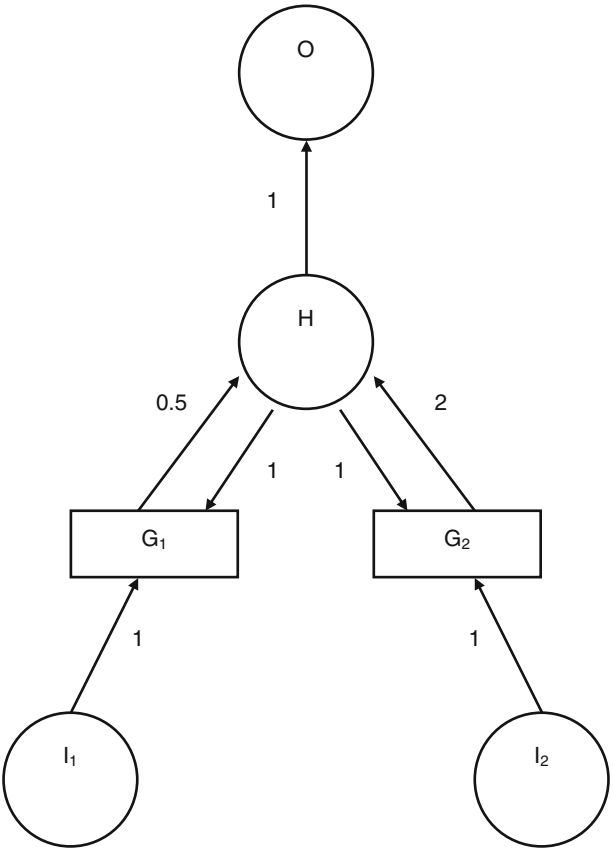

Fig. 2 A network that computes the language $a^n b^n$. Inputs are coded locally ((1,0) and (0,1) for *a* and *b*, respectively), and G_1 and G_2 are gating units, meaning that they allow a value between two nodes to pass through only if a second connection to the node (in this case, a connection from an input node) is active. The activation functions for G_1, G_2 and H are linear, and the beginning activation of node H is 0.5. The network accepts an input if the value of the output node O is 0.5, and rejects it otherwise

abcdabcd
ab(abcd)cd
ab(a(abcd)bcd)cd
ab(a(abc(a(abc(abc(ab(a(abcd)bcd)d)d)bcd)d)bcd)cd

The system must keep track of any occurrence of the basic strings that have been interrupted with an embedded instance, because every instance must complete. This means the system must track dependencies that may span a great distance across strings (as in the fifth example, above).

The structure of the Sierpinski triangle allows the system to do this by encoding symbols as points in state space. The overall space is divided into three regions, each corresponding to a symbol (in this case *a*, *b*, and *c*). Because of the fractal structure of this space, each of these regions recapitulates the structure of the whole triangle, including the symbol-specific regions themselves. This recursive spatial

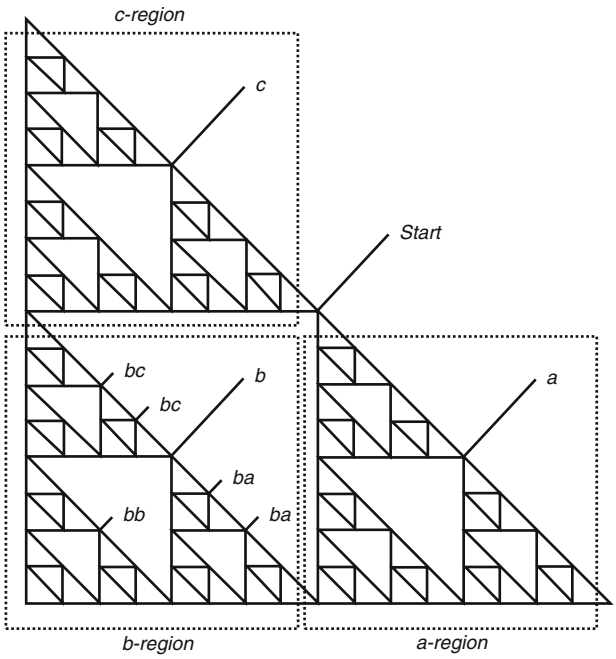

Fig. 3 The Sierpinski triangle state space (adapted from Tabor 2000)

Table 1 Tabor's (2000) center-embedded language

$S \rightarrow A\ B\ C\ D$	$A \rightarrow a\ S$	$B \rightarrow b\ S$	$C \rightarrow c\ S$	$D \rightarrow d\ S$
$S \rightarrow e$	$A \rightarrow a$	$B \rightarrow b$	$C \rightarrow c$	$D \rightarrow d$
			$C \rightarrow a\ S$	
			$C \rightarrow a$	

structure allows the space to store arbitrarily long strings. So, for example, the string *abcabc* is represented by a point in the *c*-region of the *b*-region of the *a*-region of the *c*-region of the *b*-region of the *a*-region.

The system computes by beginning at (0.5, 0.5), and updates its state according to a set of four transformation rules, selected according to the current input and network state (Tabor 2000, p. 45). The system returns to the start position iff the input string is a member of the language. Viewed across time, the system moves through state space in various cyclical trajectories. For example, to recognize the string *abcd*, the system begins in the start state. When presented with *a*, it moves to the location labeled *a* in Fig. 3, and cycles through the different regions—locations *b* and *c*—as subsequent inputs are processed. When *d* is presented, the system returns to the start state.

To handle center-embedded clauses, the system capitalizes on the self-similarity of state space, descending into nested regions. Take, for instance, the string *ababcdcd*. As before, the system first proceeds to the *a*-region and then the *b*-region

(points *a* and *b* in Fig. 3, respectively). At this point the system encounters the beginning of an embedded clause in the form a second occurrence of *a*, and thus descends a level, to the *a*-region of the *b*-region (point *ba* in Fig. 3). Processing proceeds in this sub-region, transitioning through points *bb* and *bc* before ascending back to point *b* when the first instance of *d* is encountered. The system then completes the processing of the containing string as before.

Taken in tandem with the research summarized in "Regular Languages", Moore's and Tabor's constructions put pressure on opponents of the TLS. Specifically, as previously noted, finite-node recurrent network architectures compute regular languages by capturing the structure of those languages in state space (i.e., their minimal DFA), and this is a logical consequence of their architecture (Casey 1996; Omlin and Giles 1996). Moore's and Tabor's constructions demonstrate that the strategy exhibited for regular languages can be extended to context-free languages, through the use of more complicated internal dynamics that recapitulate the constituent structure of those languages. Of course, in contrast to Casey's demonstration that a certain type of structure is necessary for computing a language, Moore's and Tabor's networks are 'proof of concept' or 'how possible' models. Consequently, they do not show that all finite-node networks must recognize non-regular languages using this method, or that networks trained on non-regular languages will typically adopt this strategy; there remains the possibility of alternatives. But given this emerging conception of language computation in networks, these results pressure opponents of the TLS to provide similarly well-developed examples of these presumed alternatives, i.e., of finite-node recurrent networks that can solve the problem *without* appealing to such structure. Furthermore, as I will now briefly argue, there is some qualitative evidence that networks trained on non-regular languages do adopt the structured state space strategy, providing further reason to favor the TLS.

Further Indicators

An early (and much-discussed) attempt at training recurrent networks to processes linguistic structure is reported in Elman (1991, 1995). An analysis of the state space of those networks revealed that the state space had reorganized along syntactic and semantic dimensions. Tabor notes that his networks are qualitatively similar to Elman's empirically observed results:

> Elman … found that different lexical classes corresponded to different subregions of the [state] space. Likewise, in the example above, the three lexical classes A, B, and C correspond to three distinct regions of the representation space… Elman also noted that the [network] followed similarly shaped trajectories from region to region whenever it was processing a phrase of a particular type, with slight displacements differentiating successive levels of embedding. Here, the single phrase *S* [i.e., one of the basic strings] is also associated with a characteristic (triangular) trajectory wherever it occurs and slight displacements also differentiate successive levels of embedding. (Tabor 2000, p. 45)

The similarity between Elman and Tabor's networks suggests that to handle syntactic information, Elman's networks settled on a state space with a structure of the same general category as that of Tabor's dynamical automata and Moore's Cantor set system.

Bodén and Wiles (2000) compared the dynamics of networks trained to recognize a context-free language ($a^n b^n$) and a context-sensitive language ($a^n b^n c^n$). Unlike the languages considered by Tabor, context-sensitive languages cannot be computed using a PDA (instead they require a form of Turing machine), so one question is whether their dynamics are of the same general sort as Tabor's and Elman's networks. Furthermore, the networks they considered were of a different type than those used by Elman and others—sequential cascaded networks (SCN)—so a second question is whether the network type has an impact on the sort of dynamics used to solve the problem.

Bodén and Wiles discovered that "the type of dynamics found … extends naturally to a language of a different complexity class and also to a different network type" (p. 208). These dynamics are at least qualitatively similar to those of discovered by Tabor and Elman:

> The SCN … divides the state space into smaller and smaller regions counting up letters. When counting down, each step is proportionally larger. The additional difficulty with $a^n b^n c^n$ [i.e., the context-sensitive language] compared with the context-free language $a^n b^n$ consists of counting up and counting down simultaneously. The network solves this by oscillating in two principal dimensions, one for counting up and one for counting down. (p. 207)

Putting the pieces together, I suggest that this research—variously involving proof, construction, and empirical observation—paints the following picture: To compute a language, finite-node networks make use of sufficiently structured state space. In the networks surveyed here, this structure is self-similar, or approximately so.[13] Furthermore, this structure captures the constituent structure of language insofar as it encodes the various combinations of atomic symbols, and operations are sensitive to those combinations. Consequently, this research challenges Clark's rejection of the TLS.

[13] I am not claiming that a self-similar, fractal structure is *necessary*, only that *some* significant structure is. This structure need not be stable. So, for example, liquid state machines (LSMs) are not counterexamples to my argument (Maass et al. 2002, 2007). In LSMs, the connectivity between nodes in the reservoir cannot be too dense, for otherwise noise can swamp out useful information. This indicates that, while it is indeed the case that the reservoir state space in LSMs are not structured specifically for the purposes of computing a particular language, they nonetheless have structure available for the encoding and retention of linguistic information. Furthermore, it appears that to get computational equivalence with traditional automata, the basic LSM architecture must be augmented with recurrent connections from readout units so that additional attractors are available, thereby providing additional stability in storing information; in other words, the space has to be made more like that of the networks described above. But, again, the thesis of this paper does not require that this particular type of structured space is necessary, only that some recapitulation of linguistic structure in state space is.

The TLS Revisited

The argument I've developed thus far runs something like this: (1) Language has a certain level of complexity, the processing of which requires tracking constituent structure; (2) in artificial neural networks, solving this task requires capturing it in the form of structured state space; (3) Therefore, some form of the TLS is correct-language use does require a significant, language-specific transformation of biologically basic forms of representation. The research summarized above was dedicated to motivating premise (2), and the remainder of this section focuses on its defense.

Michael Wheeler (2007) suggests that one can assess whether the internal states of networks recapitulate linguistic structure by considering whether their behavior can be understood *without* attributing such structure to those states. For example, he asks whether Elman's (1991, 1995) networks can be understood simply using notions such as node activation and vector transformation, i.e., without claiming that internal states track grammatical categories such as verb and noun (Wheeler 2007, p. 304). Wheeler concludes that such non-linguistic explanations fail to explain how the networks process *linguistic* information, and hence the internal states must be understood as recapitulating the grammatical categories of natural language. This reasoning would appear to apply to Tabor's network as well, since to understand how the network recognizes inputs, we must take the internal states to encode constituent structure.[14]

This same reasoning suggests that the processes operating on internal representations are language-like in the sense that they function at the level of syntactic structure. For example, appreciating how Tabor's network recognizes the string *ababcdcd* requires acknowledging that, upon encountering the first *d*, the network transitions from a state representing *ababc* to one representing *ab*—that is, the network removes three symbols from memory, leaving two. This is functionally identical to the operation of a PDA computing the language: The system removes (/pops/discards/deletes/) three items from memory, leaving two. The *way* in which the two systems go about adding and removing items from memory is of course different—they use different primitive components and operation—but *what* they do is essentially the same: Operate on linguistically structured representations by being sensitive to that structure. It is by viewing the states of the network as having constituent structure *and* by taking network operations to be sensitive to that structure that we are able to grasp how the system distinguishes between strings that are members of the language and those that are not. If this is correct, then attempts to avoid the TLS by appealing to a distinction between content and vehicle fail in this particular case. The case of language processing is importantly different than the case of color: Whereas it is the case that greenness can be represented without a green vehicle, it is not the case that linguistic structure is represented without a structured representation.

One possible response to this argument is to challenge the relevance of the evidence presented in its favor. In particular, the surveyed networks are biologically

[14] For the remainder of this paper I focus discussion on Tabor's network.

unrealistic in a variety of ways: Their components bear only an abstract resemblance to biological neurons, and they are configured by hand or through the operation of unrealistic learning procedures, for instance. Given this lack of realism, what reason is there to think that the results generalize to actual biological systems?

On the one hand, the ultimate test of external validity for these models is empirical. However, the fact that they arrive at a consensus regarding the need for structured representations provides some *prima facie* support for the claim that such structure is a very real possibility in networks with broadly similar computational characteristics and task demands. The possibility of structure should thus be taken seriously. On the other hand, the problem of ecological validity is an issue for the opponents of the TLS as well, since their positions are based on appeals to the very same types of network architectures as surveyed above. And this is the crux of my argument: Rather than refuting it, those very architectures actually support a version of the TLS.

This suggests a second possible objection: Agree that the network representations are linguistically structured, but claim this is not the *right kind* of structure. The strategy is thus to articulate a conception of what qualifies as genuine linguistic structure, and then argue that the network representations fail to satisfy this criterion. What might be the 'right kind' of structure? A reasonable place to look is in classical automata, since they form the basis for traditional philosophical debates over the format of mental representation. In this case, the objection claims that what people have in mind when they discuss constituent structure is that sort of constituency exhibited by classical automata such as Turing machines. Specifically, in such systems symbols are defined so that *instances are of the same symbol type iff they share the same physical form*. So, for example, Turing (1936) defines symbol types as topological arrangements of points on a two-dimensional plane, so that any and all instances of that symbol share the same arrangement of points (Turing 1936, p. 135, fn.). This type of constituency—call it 'Turing-constituency'[15]—is shared by natural language, as well. But networks do not compute using Turing-constituency. For example, in Tabor's network, occurrences of *a* in *bca* and *abc* have two completely different forms—if indeed one can even distinguish the form of a token *a* from that of other symbols in either case. The argument may thus show that there is *some* kind of constituency in network representations; but it is not the *right* kind of constituency; therefore, the network states do not recapitulate the structure of language.

There are at least two reasons this objection is unfounded. First, there is no a priori reason to assume that all forms of the TLS involve Turing-constituency. Indeed, Turing-constituency is much too strict when applied to biological systems: There is simply too much physical variation across individuals at the same time, within the same individual at different times, and within the same individual at the same time. To interpret the TLS as necessarily involving Turing-constituency would be to render it an extremely unlikely thesis from the start, and this suggests that

[15] Van Gelder (1990) christens this form of constituency 'concatenative' (1990, p. 360), and then promotes it to 'syntactic' (1990, p. 361).

doing so would be to misinterpret it.[16] Second, a shared background assumption in the present discussion is that mental representations are broadly connectionist (Clark 2004; Wheeler 2004, 2007). It has been pointed out many times that connectionist representations do not use Turing-constituency (often in terms of 'context sensitivity'), a claim that Clark has endorsed in the past (Clark 1993). The issue is thus not whether networks make use of Turing-constituency—they do not—but rather whether the type of constituency they do use recapitulates the structure of language.

More generally, Turing-constituency is only one of a set of equivalent methods for capturing the constituent structure of external language; an alternative being that method employed by finite-node neural networks. A similar point has been made before: Van Gelder (1990) distinguishes between two types of constituent structure: 'concatenative' and 'functional'. Concatenative constituency is Turing-constituency. Functional constituency, in contrast, "is obtained when there are general, effective, and reliable processes for (a) producing an expression given its constituents, and (b) decomposing the expression back into those constituents." (van Gelder 1990, p. 361) So Turing-constituency is a type of functional constituency, but not the only one—the constituency on display in Tabor's network is also a form of functional constituency.

This observation introduces a third possible response, a move I'd like to endorse: Since the TLS is not inconsistent with EEC, one could *concede* the TLS while retaining a trio of additional substantive theses: (1) Turing-constituency is not the appropriate type of constituency for understanding cognition (except, perhaps, when language is functioning as external scaffolding); (2) Language *qua* external structure sometimes plays an enabling role in cognition (namely, as external scaffolding); and (3) The warehouse of internal representations may have a smaller inventory than previously thought. The first thesis is motivated by the fact Turing-constituency is only one way that computational systems recapitulate constituent structure, while the second and third theses are motivated by the sorts of empirical evidence cited by Clark and described above. Consequently, this sort of position has the virtue of accommodating, at least as a methodological starting point, the various evidence we've encountered so far.[17]

[16] Van Gelder (1990) attributes the claim that cognition involves Turing-constituency to Fodor and Pylyshyn (1988). However, Fodor and Pylyshyn also sometimes identify realization with *simulation* (Hopcroft and Ullman 1979). Since simulation allows for the possibility that a realizing system may not actually use that particular method for implementing constituent structure, Fodor and Pylyshyn can presumably avoid this issue.

[17] This sort of move may be available to Clark if he were to give up his more recent rejections of the TLS. For example, in his (1993) Clark endorses Van Gelder's broadening of the notion of constituency (1993, p. 125), at least for some explanatory purposes. If he allows that linguistic structure need not involve Turing-constituency, and conceded that there is a significant recapitulation of constituent structure in mental representation in a system that lacks Turing-constituency, he could still hold that this internal structure plays little or no role outside of underwriting language use (thereby preserving the externality of language as regards other types of cognitive process). Again, this position does not seem consistent with his (2008).

Extending the Argument

The focus thus far has been on Clark's position. But, as noted in the introduction, there are others who have put similar pressures on the TLS. For example, Rowlands (1999) singles out the TLS as problematic, and attempts to undermine its motivation by sketching an alternative based on research on connectionist networks.[18] Beginning with the observation that networks are naturally suited for solving pattern-matching and pattern-completion tasks (rather than rule-governed inference), Rowlands proposes that behaviors that initially appear to require structured internal representations, "can be reduced to an *internal* process of pattern recognition and completion together with a process of manipulation of *external* … structures." (1999, p. 164) In support of this claim, Rowlands discusses several examples, including Rumelhart et al.'s (1986) proposal for how connectionist networks might perform multiplication, and Bechtel and Abrahamsen's (1991/2002) discussion of how logical inference may proceed in connectionist networks. In the former, Rumelhart et al. speculate that networks may perform multiplication by manipulating the environment so that patterns can be discovered by internal processes (which do not contain constituently structured representations). So, for example, when multiplying large numbers, we arrange the numerals on some external medium (e.g., paper) in such a way that the digits form columns. Then, internal pattern-matching processes detect patterns in the arrangement of numerals, filling in gaps. So, for example, if a '2' is above a '4', the missing component to the pattern is '8'. The external environment is then adjusted by adding the new numeral to the page, revealing new patterns for detection and completion. In this case, the argument goes, the need for structured internal representations has been off-loaded into the environment.

The results of "Language and Biologically Basic Representation" suggest that we treat this argument with caution. In light of those results, we should expect a network that performs multiplication or logical transformations to also recapitulate constituent structure, *if the network has to track that structure*. One way to interpret Rowlands' central claim, then, is that performing multiplication in the way just described is precisely a case where the network has been relieved of any duty for tracking internal structure—that task is moved to the environment. But this is not obviously true. For example, a network that performs multiplication by manipulating external symbols must be able to *parse* a string of numerals into its constituents so as to perform the pattern-completion task; the system is causally sensitive to changes in that internal structure, e.g., '657' versus '667'. Furthermore, the network must encode information about the structure of the workspace in the environment in order to place new numerals in the correct location (e.g., during carryover). Consequently, it is not clear that putting structured representations into the environment obviates the need for a network to track constituent structure, and if it still needs to track the constituents of external representations, then the research

[18] The view challenged by Rowlands is: "If thought or language use has a certain structure, a structure judged to be essential to it as such, then the internalist picture tempts us into believing that the relevant processes occurring inside the head, processes which allow us to think or use language, must have that same structure." (1999, p. 148) Rowlands' 'internalist' target qualifies as version of the TLS as described in "Introduction."

summarized in "Language and Biologically Basic Representation" suggests that the structure will be internally recapitulated.

More generally, as noted at the beginning of "Language and Biologically Basic Representation", it is not a counterexample to the TLS to show that networks not tasked with tracking constituent structure do not internally recapitulate that structure. Rather, the key issue is what happens when networks must be causally sensitive to constituent structure, internally or externally. So, what must be shown is that a system tasked with tracking externally instantiated constituent structure can avoid internal recapitulation through the use of some other set of (alternatively structured) processes. On the one hand, if the multiplication network does not need to parse external representations into their constituents, then it is not a case where we would expect any pressure for internal recapitulation; on the other, if, as just argued, it does track constituent structure, then the research appealed to in "Language and Biologically Basic Representation" suggests that the network will reflect that structure in its internal dynamics.

A similar response can be given to Rowlands' appeal to Bechtel's (1994) logical derivation network.[19] Roughly, the network has eight input 'slots' (i.e., clusters of input nodes), each of which contains a vector encoding of a logical formula (e.g., 'A & B'). One of these slots contains the desired conclusion of a proof, three contain premises, and the remainder will hold intermediate steps in the derivation. The network is initially presented with the desired conclusion and the premises, and it is charged with outputting the first intermediate derivation. The outputted formula is then added to the set of inputs, and the network is tasked with producing the second intermediate inference in the proof. This continues until all eight input slots are filled, and the last output generated is the conclusion.

In this case what we find is that the network is not a case of a system being causally sensitive to external constituent structure without internal recapitulation, because it is not tasked with being sensitive to that structure in the first place. For instance, the network is not asked to recognize logical form in general, such as the fact that (A & B) and ((D $\supset$ G) & $\sim$(X & $\sim$Y)) are instances of the same general form—*p* & *q*—and that the second contains a sub-formula that itself is an instance of that form. More importantly, the network is not responsible for parsing the inputs it *does* receive, into premises, intermediate conclusions, etc.—all of this work is taken care of by the processes responsible for presenting inputs to the network. Imagine, for example, the network is working with a pen and paper, reading formulas off the paper and writing new derivations in a way similar to the multiplication example previously considered. In the current simulation, it's as if there is 'pre-processor' responsible for parsing inputs from the paper and assigning them to discrete slots in the network, and a 'post-processor' for taking network outputs and writing them in the correct location on the paper; the network never has to do any significant parsing of input or output strings. Again, additional argument is required to establish that this network is a genuine counterexample to the TLS.

[19] The description given here is based on the summary given in Bechtel and Abrahamsen (1991/2002, pp. 115–117).

Notice that in challenging Rowlands' appeals to the multiplication and logical inference networks I am not rejecting the possibility that much of what goes on in performing multiplication or logical inference involves pattern matching or, more generally, systems of internal representation that are less complex than those needed to track constituent structure. Rather, the point is that we cannot off-load *all* tracking of structure to the environment, because capitalizing on the affordances offered by external linguistic structures often involves tracking constituents of those structures. Consequently, the position advanced here is consistent with the view that there is a plurality of internal representational schemes, some of which robustly recapitulate linguistic structure, while others do not.[20]

A different sort of challenge to the TLS is given by Michael Wheeler (2004, 2007), who adopts a moderate position by allowing for a partial recapitulation of linguistic structure. Clark proposes that linguistically competent individuals can engage in a form of 'self-directed speech' wherein missing external linguistic elements are accommodated for by internal surrogates representing those missing elements. (The conceptual matching task described in "The EEC Case Against Linguistic Structuring" is an example of this capacity in action.) Despite representing linguistic structure, Clark claims, they do not recapitulate that structure (Wheeler 2007, p. 301). Wheeler argues that allowing for internal surrogates for external structure is tantamount to conceding that those surrogates recapitulate that structure. In cases where the necessary linguistic structures are present in the environment, relevant cognitive processes are "directly causally locked" onto those features. In off-line reasoning, at least some of these structural features, essential to enabling the completion of the task, are by assumption *missing* from the environment. But, Wheeler notes, if what is important for task completion is that internal processes be 'causally locked' onto structural features, and those structural features are not present externally, they must be present in the surrogate. Therefore, Wheeler concludes, "in the off-line case [of language-based reasoning] we confront nothing less than a profound transformation in the brain's own basic mode of representation…" (2004, p. 711)

There are two important additions that must be noted. First, Wheeler's argument does not establish *which* structural features must be recapitulated. In an addendum to his original paper, Wheeler (2007) appeals to Elman's (1995) much-discussed case of connectionist language processing ("Further Indicators"), arguing that the appropriate way to interpret their internal states is as reflecting the structure of linguistic categories. Recall that the state spaces of these networks include regions corresponding to different semantic and syntactic categories (e.g., nouns cluster in one area of state space, while verbs group in another). For Wheeler, this is sufficient to conclude that linguistic structure is present in the representational system

[20] So, for example, Bechtel and Abrahamsen (1991/2002) argue that students learning logic appear to engage in pattern-recognition rather than the acquisition of abstract rules. Nothing in my argument is inconsistent with this claim. Perhaps there is a low-level parsing mechanism in place that allows students to individuate formulas and their constituent symbols, and the outputs of this mechanism are fed to a different system that uses a pattern-completion strategy. Mastering logical syntax is difficult precisely because this default strategy needs to be overcome, either by reconfiguring the dynamics of one or the other networks, or by redirecting information flow to a representational system that has the requisite internal structure.

(Wheeler 2007, p. 304). So, the first addition is that internal representations will recapitulate at least some parts of the structure of the grammar being processed, such as distinctions between grammatical categories.

The second addition is a proposal for minimizing the extent of this recapitulation. Wheeler claims that even though representations are structured, the processes which manipulate them need not be 'language-like'—instead they are typical connectionist pattern-matching and pattern-completing operations (2004, p. 711). So, for example, Wheeler holds that the representations "recapitulate the structural properties of natural language," but "[t]he mechanisms which deal in the linguistically structured representations may themselves be domain-general in character (e.g., generic connectionist pattern-completers)." (p. 712) To sum, according to Wheeler, (1) the capacity to engage in self-directed speech establishes the need for structured internal representations, (2) representations will recapitulate at least the structure of grammatical categories, and (3) despite representations recapitulating structure, the processes that operate on them do not.

There are several disagreements between Wheeler's position and the one advanced in "Language and Biologically Basic Representation". The most obvious is that the full-scale recapitulation of constituent structure advocated above is stronger than Wheeler's, since it posits more than the mere replication of grammatical categories. However, this difference could be merely cosmetic. As noted above, in his (2007) addendum, Wheeler adopts a strategy of looking at networks that process language (namely, Elman's networks) to see if their internal representations are linguistically structured. Given that the methodology of "Language and Biologically Basic Representation" is similar, Wheeler may simply allow for additional linguistic structure in internal representation when presented with additional models to scrutinize.

It is not as easy to dispel the second point of departure, namely, that operations over structured representations need not themselves be 'language-specific' but rather 'domain-general'. This proposal can be interpreted in different ways. One possibility is that the processes operating over linguistic representations in connectionist networks are 'domain general' in the sense that the same types of processes are active when networks are engaged in non-linguistic tasks—all operations are vector transformations, for example. This interpretation seems too broad, however, since, as we have seen, networks can be just as 'linguistic' as Turing machines, even though they work with vectors. A second interpretation is that, while the representations are structured, the processes operating over them do not pay attention to that fact: Operations over linguistically structured representations in connectionist networks are not causally sensitive to that structure. This proposal seems too strong. First, processing in Elman's network is clearly causally sensitive to linguistic category—a representation of a noun will be processed differently than a representation of a verb, as reflected by the fact the two types of grammatical category are represented in different regions of state space. So the proposal does not cohere with Wheeler's (2007) addendum. Second, as argued in "Language and Biologically Basic Representation", networks charged with processing language *are* causally sensitive to constituent structure. So being causally sensitive to structure is part and parcel with the recapitulation of structure. Finally, on this interpretation it simply is hard to

see *why* internal representations would bother to reiterate linguistic structure if doing so played no role in processing; indeed, it seems to go against the principles of embedded and embodied cognition as endorsed by Wheeler.

Perhaps there is a third possibility: Distinguish between two kinds of connectionist information processing, 'pattern-completion' and 'linguistic-manipulation', and argue that the latter is not involved in human mental representation, even if many artificial networks happen to use the strategy. In this case the burden of proof is on those who would endorse such a move. The challenge is to (1) articulate a principled distinction between pattern-completion and linguistic-manipulation operations in connectionist networks, and (2) demonstrate that the former, taken alone, is sufficient to underwrite processes capable of handling at least context-free grammars. If the methodology is to consult actual networks that compute languages, then it is hard to see how this challenge can be met.

Conclusion: Internal Scaffolding

Against those who would reject the TLS, I have argued that research on formal language processing in artificial neural networks indicates that internal representations will robustly recapitulate the constituent structure of language. However, nothing in my argument is inconsistent with EEC as an approach to understanding cognitive behavior; instead, it works in tandem with the embedded and embodied methodology.

Take, for example, the proposal to let the world be its own model. The motivation behind moving representations into the environment is not for the sake of getting rid of representations; rather, the purpose is to enable a system to complete tasks it otherwise has difficulty doing. This may still be a relevant consideration even if the system recapitulates a significant amount of linguistic structure, because the *way* in which a system encodes structure imposes inherent constraints on how that structure is processed. Having an alternative encoding of that structure in the environment may thus provide the key to overcoming the intrinsic limitations of the on-board representational system. Consider, for instance, the representational scheme in Tabor's context-free language network. In that network, keeping track of center-embedded clauses requires 'descending' to a lower-level reiteration of the Sierpinski triangle. As successive embedded clauses are encountered, the network continues descending into finer grained sub-spaces. Given that biological information processing systems are subject to background noise, we can imagine a case where noise is injected into Tabor's network. This noise should have a disproportionate effect on the network's ability to track nested center-embedded clauses, since the network must make finer-grained distinctions the deeper a clause is embedded. This does not happen for left- or right-branching sentences, so the effect of background noise should be negligible in those cases: When subjected to noise, the networks should have more difficulty parsing center-embedded strings when compared to left- or right-branching strings.

Aside from the fact this prediction is qualitatively similar to human behavior, it suggests a role for external symbols: Assisting the on-board representational system

with the tracking of center-embedded clauses, perhaps through the reordering of constituents into a form that is easier for the internal system to parse. Suppose, for instance, we are given a center-embedded sentence:

The dog the boy the man the police officer arrested scolded chased bit the girl.

In this situation our natural tendency is to fiddle around with the written words, perhaps by extracting and reordering them or by drawing lines to connect related clauses. So it may be that the external symbols assist in overcoming a constraint imposed by the implementation details of the on-board representational system. This example illustrates that significant recapitulation of linguistic structure does not mean the EEC approach is warrantless. Instead, *understanding how the internal mechanism handles such structure can actually suggest ways in which environmental props factor into cognitive processes*.

Such an understanding can also suggest new internal mechanisms that operate on principles similar to those favored by opponents of the TLS. Take, for example, Clark's various accounts of the causal role of external language:

> The computational value of a public system of essentially context-free, arbitrary symbols, lies … in the way such a system can push, pull, tweak, cajole, and eventually cooperate with various nonarbitrary, modality-rich, context-sensitive forms of biologically basic encoding. (2008, p. 47)
>
> Encounters with words and with structured linguistic encodings act to anchor and discipline intrinsically fluid and context-sensitive modes of thought and reason. (2008, p. 53)
>
> Words and sentences act as artificial input signals … that nudge fluid natural systems of encoding and representation along reliable and useful trajectories. (2008, p. 53)

According to Clark, the basic idea is that linguistic structures guide non-linguistic connectionist representations in ways those representations could not guide themselves. If my argument is correct, we may now also draw on linguistically restructured networks when speculating on the computational nature of cognition. In particular, when we put together linguistically restructured networks and 'biologically prior' networks, we get hybrid systems whose functionality is consistent with Clark's description of the role of external linguistic structures. In such systems linguistic scaffolding is not only external, but also internal. To motivate this perspective, I offer a simple mechanistic sketch based on the standard XOR network.

Figure 4 depicts a common XOR network. An interesting feature of this network is the contribution made by the initial set of connections with weight -1 (the 'negative connections'): If these weights are changed to zero, the network computes OR rather than XOR. So, if there were a way to change those weights on the fly, the network could be used to compute different functions at different times.

Suppose we have a network that realizes a two-state DFA, as described above. Because such a network will have a structured state space, the DFA network can be interfaced with the XOR network using inter-network connections. These connections emanate from nodes of the DFA network and terminate on the weights of the XOR network, and their activity is such that they inhibit the negative connections iff

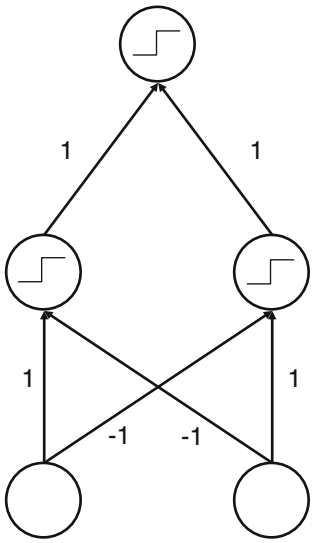

Fig. 4 An XOR network

the activity of the DFA network places it in a certain region of its state space. That is, when the DFA network is in one region of state space, the intermediate connections allow the negative connections of the XOR network to do their job, but when the DFA network is in the other region of state space, the negative connections are inhibited. The result is a network that switches between computing two different functions depending on the state of the controlling DFA network.

We can also view the actions of the DFA network as transformations of the state space topology of the XOR/OR network. When computing OR, the first layer of weights simply sends the initial inputs to the hidden units without modification. In contrast, when computing XOR the first layer of weights transforms the plane of input values (of which we only care about (0,0), (0,1), (1,0), and (1,1)) into a line with slope -1. If we look at the relationship between these two modes by incrementally modifying the contribution of the inhibitory connections (from 0 to -1) and observe the effects on hidden unit space, we see that the controlling network distends state space by collapsing the plane (used to compute OR) into a line (used to compute XOR). In metaphorical terms, it's as if one grasped the opposite corners of a rubber square and pulled outwards—the free corners move towards the center of the square. In this way, the structured state space of the DFA network can be viewed as an *anchor* that *pulls* and *cajoles* the space of the XOR/OR network into new configurations, allowing different functions to be computed.

More generally, if the main argument of this paper is sound, networks responsible for linguistic computation will have complex but structured state spaces and transitions between regions of those spaces. The proposal, then, is that such systems serve as *anchoring* networks, interfacing with other networks whose representational space is structured non-linguistically—perhaps along semantic dimensions (Clark 1993, chapter 6; Churchland 2007). This interface allows the semantic state space to be restructured on the fly in ways that could not be achieved without the

assistance of the anchoring network, which amounts to enabling the semantic network to manifest previously unattainable behavior.

For example, consider the Hermer-Vazquez et al. (1999) conjoined cue study, discussed by Clark (2008). In that study, pre-linguistic subjects were unable to quickly locate a hidden object on the basis of environmental cues because they only tracked one location-identifying feature (wall geometry or color), while the conjunction of two such features were necessary to uniquely determine the object's location. It's as if pre-linguistic subjects chose one environmental feature to track, while discarding the other. Linguistically competent subjects, on the other hand, were capable of keeping both environmental cues in mind, and hence were able to locate the hidden object on the first attempt. On Clark's account, linguistically competent subjects are able to solve the task because they token an *unstructured* internal state that represents the constituent structure of an external linguistic entity; pre-linguistic subjects thus fail because they lack the requisite unstructured internal states. On the alternative suggested here, pre-linguistic subjects fail, despite their ability to track environmental features such as wall color and geometry, because they are unable to bring the regions of semantic space underlying those individual abilities into proximity with each other: Pre-linguistic subjects have the necessary semantic spaces, but they lack the means by which to coordinate regions of those spaces. Linguistically competent individuals, in contrast, are able to solve the task because they have a linguistically structured anchoring space that tweaks and prods the previously established semantic space in principled ways. The semantic space can be transformed so that wall color and geometry are simultaneously relevant because the anchoring space has a constituent structure that brings syntactic items into functional proximity, dragging the associated semantic regions along. On Clark's account, linguistically competent subjects have the means by which to represent the required *external* scaffolding; on the proposed alternative, they have the requisite *inner* scaffolding.

This of course is only a sketch of a possible mechanism, and I provide no empirical assessment of its plausibility. But it nonetheless suggests a complementary alternative to the view of language advanced by Clark. Situated approaches to language may discount the impact of language on internal representational systems, opting instead to keep linguistic structure in the world. By keeping such structure in the world, language can function as a scaffold for enabling biologically basic forms of internal representation to function in new ways while not affecting in any fundamental way the structure of those representations. Results from research on how artificial neural networks compute language suggests that this view is untenable: There must be a significant degree of representational restructuring to handle language processing. The TLS is thus vindicated. However, this conclusion does not threaten the claim that language can be profitably viewed from the perspective of situated cognition: The thesis is compatible with the idea that language serves as external scaffolding for thought. But this external scaffold is not the only way in which biologically basic forms of representation are subverted into performing new tasks. The illuminating metaphor of language as a set of anchors, enabling the stable shifting of other, non-linguistic forms of representation, remains in play. The issue is over the location of the scaffolding: Clark (1998) stipulates that

scaffolding is "external support" (p. 169), but the term is actually more suggestive. During the construction of a building, scaffolding is erected both outside and *inside* the building, and this I propose is also a potentially useful way to view language.

References

Bechtel, W. (1994). Natural deduction in connectionist systems. *Synthese, 101*, 433–463.

Bechtel, W. & Abrahamsen, A. (1991/2002). *Connectionism and the mind*. Oxford, UK: Blackwell. Page references refer to the (2002) second edition.

Bodén, M., & Wiles, J. (2000). Context-free and context-sensitive dynamics in recurrent neural networks. *Connection Science, 12*(3/4), 197–210.

Bodén, M., Wiles, J., Tonkes, B., & Blair, A. (1999). Learning to predict a context-free langauge: Analysis of dynamics in recurrent hidden units. *Ninth International Conference on Artificial Neural Networks* (*ICANN99*) (pp. 359–364). Edinburgh, Scotland: IEEE Press.

Boysen, S., Berntson, G., Hannan, M., & Cacioppo, J. (1996). Quantity-based interference and symbolic representations in chimpanzees (pan troglodytes). *Journal of Experimental Psychology, 22*(1), 76–86.

Brooks, R. (1991). Intelligence without representation. *Artificial Intelligence, 47*, 139–159.

Casey, M. P. (1996). The dynamics of discrete-time computation with application to recurrent neural networks and finite state machine extraction. *Neural Computation, 8*(6), 1135–1178.

Chomsky, N. (1963). Formal properties of grammars. In R. D. Luce, R. R. Bush, & E. Galanter (Eds.), *Handbook of mathematical psychology* (Vol. 2, pp. 323–418). New York: Wiley.

Christiansen, M. H., & Chater, N. (1999). Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science, 23*, 157–205.

Churchland, P. M. (2007). *Neurophilosophy at work*. Cambridge, UK: Cambridge University Press.

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

Clark, A. (1998). Magic words: How language augments human cognition. In P. Carruthers & J. Boucher (Eds.), *Language and thought: Interdisciplinary themes* (pp. 162–183). Cambridge: Cambridge University Press.

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, UK: 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., & Tviskin, S. (1999). Sources of mathematical thinking: Behavioral and brain imaging evidence. *Science, 284*, 970–974.

Dennett, D. (1991). *Consciousness explained*. Boston: Little, Brown & Co.

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning, 7*, 195–225.

Elman, J. L. (1995). Language as a dynamical system. In R. Port & T. Van Gelder (Eds.), *Mind in motion: Explorations in the dynamics of cognition* (pp. 195–225). Cambridge, MA: MIT Press.

Fodor, J. A. (1975). *The language of thought*. Cambridge, MA: Harvard University Press.

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

Hermer-Vazquez, L., Spelke, E., & Katsnelson, A. (1999). Sources of flexibility in human cognition: Dual-task studies of space and language. *Cognitive Psychology, 39*, 3–36.

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

Maass, W., Joshi, P., & Sontag, E. D. (2007). Computational aspects of feedback in neural circuits. *PLoS Computational Biology, 3*(1), 15–34.

Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation, 14*, 2531–2560.

Marcus, G. F. (1999). Connectionism: With or without rules? Response to J.L. McClelland and D.C. Plaut. *Trends in Cognitive Sciences, 3*(5), 168–170.
Marcus, G. F., Vijayan, S., Bandi Rao, S., & Vishton, P. M. (1999). Rule learning by seven-month-old infants. *Science, 283*, 77–88.
Moore, C. (1998). Dynamical recognizers: Real-time language recognition by analog computers. *Theoretical Computer Science, 201*, 99–136.
Omlin, C., & Giles, C. (1996). Extraction of rules from discrete-time recurrent neural networks. *Neural Networks, 9*(1), 41–52.
Plunkett, K., Hu, J., & Cohen, L. (2008). Labels can override perceptual categories in early infancy. *Cognition, 106*, 665–681.
Pullum, G., & Gazdar, D. (1982). Natural languages and context-free languages. *Linguistics and Philosophy, 4*(4), 471–504.
Rodriguez, P., Wiles, J., & Elman, J. (1999). A recurrent neural network that learns to count. *Connection Science, 11*(1), 5–40.
Rowlands, M. (1999). *The body in mind: Understanding cognitive processes*. Cambridge, UK: Cambridge University Press.
Rowlands, M. (2009). Situated representation. In P. Robbins & M. Aydede (Eds.), *The Cambridge handbook of situated cognition* (pp. 117–133). Cambridge, UK: Cambridge University Press.
Shieber, S. M. (1985). Evidence against the context-freeness of natural language. *Linguistics and Philosophy, 8*(3), 333–343.
Siegelmann, H. (1999). *Neural networks and analog computation: Beyond the turing limit*. Boston, MA: Birkhäuser.
Spivey, M., & Richardson, D. (2009). Language processing embodied and embedded. In P. Robbins & M. Aydede (Eds.), *The Cambridge handbook of situated cognition* (pp. 382–400). Cambridge, UK: Cambridge University Press.
Tabor, W. (2000). Fractal encoding of context-free grammars in connectionist networks. *Expert Systems, 17*(1), 41–56.
Thompson, R., Oden, D., & Boysen, S. (1997). Language-naive chimpanzees (Pan troglodytes) judge relations between relations in a conceptual matching-to-sample task'. *Journal of Experimental Psychology, 23*(1), 31–43.
Turing, A. M. (1936). On computable numbers, with an application to the *Entscheidungsproblem.* In *Proceedings of the London mathematical society*, Vol. 42 (series 2), pp. 230–256.
Van Gelder, T. (1990). Compositionality: A connectionist variation on a classical theme. *Cognitive Science, 14*, 355–384.
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?'. In B. Wallace, A. Ross, J. Davies, & T. Anderson (Eds.), *The mind, the body and the world: Psychology after cognitivism?* (pp. 298–305). Exeter, UK: Imprint Academic.