Marr's levels and the minimalist program

Mark Johnson[1]

Abstract A simple change to a cognitive system at Marr's computational level may entail complex changes at the other levels of description of the system. The implementational level complexity of a change, rather than its computational level complexity, may be more closely related to the plausibility of a discrete evolutionary event causing that change. Thus the formal complexity of a change at the computational level may not be a good guide to the plausibility of an evolutionary event introducing that change. For example, while the Minimalist Program's Merge is a simple formal operation (Berwick & Chomsky, 2016), the computational mechanisms required to implement the language it generates (e.g., to parse the language) may be considerably more complex. This has implications for the theory of grammar: theories of grammar which involve several kinds of syntactic operations may be no less evolutionarily plausible than a theory of grammar that involves only one. A deeper understanding of human language at the algorithmic and implementational levels could strengthen Minimalist Program's account of the evolution of language.

Chomsky's Minimalist Program is largely motivated by the challenge of explaining the evolution of language (Chomsky, 1995). The relatively small difference between the genomes of humans and non-human primates, and the apparently rapid and discrete nature of the emergence of human language, suggests that a single evolutionary event, perhaps a single mutation, may be responsible. The challenge then is to explain how a small, simple change to the genome could result in the emergence of human language. The Minimalist Program posits that the essential character of human language follows from a single simple, human language specific, principle or mechanism – a recursively-applying MERGE operation – that interacts with other general, non-language-specific, cognitive mechanisms or principles to produce human language. Under this hypothesis, language evolution basically consisted of a single evolutionary event which made MERGE available, perhaps fine-tuned by later natural selection (Berwick & Chomsky, 2016). Thus evolutionary considerations motivate a theory of grammar in which the human language specific principles or mechanisms are as simple as possible, on the assumption that this makes it more plausible that a single evolutionary event could have introduced them.

This paper poses the question: what kind of simplicity is likely to be most related to the plausibility of an evolutionary event introducing a change to a cognitive system? The notion of simplicity in the Minimalist Program is simplicity of the competence grammar, i.e., the formal system that specifies the possible linguistic representations. Specifically, MERGE is a simple formal operation that yields the kinds of hierarchical structures found in human languages, and the Minimalist Program hypothesises that the evolution of human language involved a single event introducing MERGE. However, because the relationship between the genome and human language is complex and indirect, the simplicity of formal operations in the competence grammar might not have any clear relationship to the plausibility of an evolutionary event introducing them, and so might not be a good guide for identifying evolutionarily-plausible theories of language. Of course standard "Occam's razor"

✉ Mark Johnson
Mark.Johnson@MQ.edu.au

[1] Macquarie University, Computer Science, Sydney, Australia

considerations argue for theories which are as simple as possible, and Berwick and Chomsky (2016) provide a variety of linguistic and other evidence for MERGE. However the point of this paper is narrower: there's no reason to believe that the simplicity of a formal description of a cognitive system is closely related to the plausibility of an evolutionary event introducing that system.

Marr's famous "levels of description" of a cognitive system helps clarify why the complexity of a change at the computational level in a formal system may have not have much relationship to the plausibility of an evolutionary event introducing that change. Marr (1982) proposed that cognitive systems, including language, should be understood in terms of three levels of representation. The implementational level is the most concrete: it describes the system in terms of circuitry, or the hardware or wetware that instantiate the cognitive process. The algorithmic level describes a cognitive system in terms of the representations and data structures involved and the algorithms that manipulate these representations. The computational level is the most abstract: it describes the goal(s) of the system, the information that it manipulates and the constraints it must satisfy. Linguistic theories are computational-level theories of language, while psycholinguistic theories of comprehension or production are algorithmic-level descriptions of how knowledge of language can be put to use.[1]

Marr pointed out that these levels are relatively independent: often the same algorithm can be implemented either in silicon or in neural circuitry, there are often several different parsing algorithms for the same grammars, etc. The complexity of a system – or of a change to a system – can vary wildly from one Marr level to another. To take an artificial example, the allowable rules in a class of grammars determines the class of languages that the grammars can generate. As Chomsky (1959) showed, grammars with rules of the form $A \rightarrow x$ and $A \rightarrow x B$ generate finite-state languages, while grammars with rules of the form $A \rightarrow x$ and $A \rightarrow C B$ generate context-free languages (where A, B, C are nonterminals and x is a terminal). Thus a very simple formal change (the substitution of a nonterminal for a terminal) dramatically changes generative capacity.

However, algorithms for parsing and generating with context-free grammars are very different to those for finite-state grammars (Aho and Ullman, 1972). Finite state automata recognise finite-state languages, push-down automata recognise context free languages, and linear bounded automata recognise context-sensitive languages (Kuroda, 1964). These different kinds of automata require very different patterns of memory access, which may

require major implementation-level changes. For example, push-down automata require *two* operations (a push operation and a pop operation) that finite-state automata do not possess. In general, specialised hardware that can implement a finite-state acceptor will not be able to implement a context-free or a context-sensitive acceptor.

In fact, there is a hierarchy of language families that lies between the context-free and the context-sensitive families – including the so-called "mildly context-sensitive languages" – that seem more relevant to natural language (Joshi et al., 1991; Steedman, 2014). Under one formalisation, Minimalist Grammars define mildly context-sensitive languages (Michaelis, 2001). Mildly context-sensitive languages are recognised by a specialised class of automata called embedded push-down automata, which roughly consist of push-down stacks of push-down stacks (Weir, 1994). Thus the relationship between the formal properties of grammars and the associated classes of automata is quite complex, and a simple change to a formal system can change radically the class of automata that recognise the corresponding languages.

While these formal grammar results aren't necessarily indicative of the properties of human language, they do show that the complexity of a change to a system can vary dramatically at the computational, algorithmic and implementational levels. We understand the algorithmic and implementation properties of Minimalist Grammars much less well than we understand these for formal grammars, but it seems likely that Minimalist Grammars are no simpler than context-free grammars at the algorithmic and implementational levels (Stabler, 2013). Berwick and Chomsky (2016) discuss the relationship between the computational and algorithmic levels, and point out that minor changes at the algorithmic level can be complicated to describe at the computational level. In summary, a change that is very simple at the computational level can be far more complex at the other levels, so the simplicity of a computational level change is no guarantee that the associated change at other levels will also be simple.

We know very little about how the plausibility of a discrete evolutionary change (e.g., a mutation) relates to the complexity of the associated change at each of Marr's levels. But if we think of the genome as a kind of specification for the construction of an organism, it seems reasonable that complexity of genomic encoding would be most closely related to complexity at the implementational level. For example, the complexity of the instructions for building a calculator is more closely related to the complexity of its wiring diagram than it is to the complexity of the axioms of the arithmetic it implements. Of course the genome isn't a blueprint for the organism; instead it interacts in complex ways with the environment and other factors to determine the organism. A more accurate metaphor views

---

[1]Poggio (2012) suggests that learning and evolution should be viewed as additional levels at which a cognitive system can be understood; see also Fitch (2014).

the genome as a kind of recipe or program that constructs the organism in ways that can depend on the environment. But because this recipe or program must ultimately specify the implementational level of the organism, we would still expect genomic complexity to be more closely related to the implementational level than any other Marr level.

Of course all the levels of representation are relevant and important for a scientific understanding of a cognitive system. But for assessing the explanatory power of an evolutionary account the implementational level plays a special role, since it is the implementation (i.e., the neural structures that enable language) that needs to be constructed by the interaction of genetic endowment and the environment. On the other hand, computational level descriptions might not need to be independently explicitly encoded in the genome (perhaps computational descriptions are best understood as scientific theories about cognitive systems?). Returning to the calculator example, all that is required for the calculator to behave correctly is that its circuitry is wired in a certain way; the axioms of arithmetic do not need to be explicitly represented in the device. Of course those axioms may help us understand why the calculator is wired the way it is, but the complexity of those axioms doesn't affect the complexity of the instructions for building the calculator independently of the complexity of the wiring.

These observations have implications for the theory of grammar. If the relationship between the complexity of a computational level change and the plausibility of an evolutionary event causing that change is as weak as suggested, then a theory of grammar which posits several syntactic primitives may be no less evolutionarily plausible than a theory that posits only one syntactic primitive, such as the Minimalist Program. For example, Combinatory Categorial Grammar (CCG), which posits around half a dozen universal syntactic combinatory operations (Steedman, 2000), may be similiar to Minimalist Grammar in terms of complexity of genomic encoding, for all we know. CCGs define mildly context-sensitive grammars (Vijay-Shanker et al., 1987), just like Minimalist Grammars, so at the implementational level they may be very similiar (e.g., embedded push-down automata should be able to recognise both). Thus evolutionary considerations alone do not strongly support Minimalist Grammar over other theories of grammar, given our current level of understanding. Steedman (2014) goes further to argue that the algorithmic and implementation complexity of MERGE makes it unlikely that the evolution of MERGE was the event introducing human language, and proposes an alternative account of the evolution of language in which CCG's different kinds of combinatory operations are independently motivated in non-linguistic cognitive terms.

There are at least two approaches one might take to strengthen a minimalist account of language evolution. If the relationship between the complexity of a computational level change and the plausibility of an evolutionary event causing that change is as weak as suggested, then a theory of grammar which posits several syntactic primitives may be no less evolutionarily plausible than a

there are systematic reductions from the computational level to the other levels that the organism can exploit, then the genome might encode a computational-level description of human language. Then a simple change at the computational level might correspond to a simple genomic change.

For example, the Parsing as Deduction approach proposes that syntactic parsers and generators are constructed by applying general-purpose inference procedures to suitably-encoded computational-level grammars (Pereira and Warren, 1983; Johnson, 1989), so the algorithmic level is derived from the computational level by general principles. Bayesian inference and other varieties of statistical inference can also be understood as connecting the computational and algorithmic levels for tasks such as parameter setting and the acquisition of the lexicon; given a computational level description of what is to be learned, there are systematic ways of applying standard algorithms to construct a learner for that information (Goldwater et al., 2009; Johnson, 2013; Perfors et al., 2011). Perhaps the mind/brains of certain animals have general-purpose "compilers" that can map computational-level descriptions onto neural circuitry.

However, assuming such a compiler is not unproblematic. It would be strange if many animals have compilers that can implement Merge but only humans actually use for Merge, perhaps like a bird species capable of flight yet never having flown (Chomsky, 1988). Also, the computational complexity of the problems such a compiler would have to solve might be very great: e.g., for the calculator example above, it is the problem of deriving the calculator's wiring diagram from the axioms of arithmetic ((Sipser, 1997) discusses the relationship between functions and the circuits that implement them).

Positing mechanisms that relate different levels of representation is also controversial for more theoretical reasons: they seem to require the grammar and linguistic principles be explicitly represented in the brain (Chomsky, 1986). However, if accounts like these could be extended to explain how Minimalist Grammar is causally related to genomic encoding or neural circuitry, then the Minimalist explanation of language evolution would be stronger.

Another approach would be to seek a minimalist account of the evolution of language directly at the implementational level. That is, we would try to find a simple change to neural architecture that would allow it to generate human language. For example, if we could show that adding a certain kind of recurrent connection to a neural network (Elman, 1990) gave it the ability to generate human language, then perhaps human language could have evolved via a mutation that introduced such connections. Unfortunately we don't know of any specific characteristic of a neural network that enables it to generate human language. Our lack of knowledge of how linguistic knowledge and

information is represented in neural circuitry makes this approach very challenging: we are only beginning to discover how hierarchical structures such as trees might be represented and manipulated in neural circuits (Smolensky, 1990; Smolensky et al., 2016), if they are represented at all (Berwick & Chomsky, 2016). To the extent we understand the relationship between network architecture and the phenomena the network can describe, it seems that multilayer neural networks are universal function approximators (Hornik, 1991), which suggests that such an approach would be challenging with our current methods.

To summarise, the plausibility of the Minimalist Program's explanation for the evolution of language depends on the plausibility that a single genomic change could be responsible for introducing language. The change that the Minimalist Program hypothesises is responsible for the evolution of human language – the introduction of MERGE – is simple at the computational level. However, this simplicity does not show on its own that the corresponding genomic change is plausible, as the change at the algorithmic and implementational levels could be quite complex. The complexity of the largely unknown implementational level changes would seem to be at least as relevant to genomic complexity as the simplicity of the change at the computational level. This suggests that theories of grammar that aren't minimal at the computational level may be as evolutionarily plausible as Minimalist Grammar. Simplicity at the computational level would be more directly connected to the plausibility of an evolutionary event if we could show systematic connections between the computational level and the other Marrian levels. Alternatively, it might be possible to provide a minimalist explanation of the evolution of human language if we could identify a simple change at the implementational level that generates human language. In any event, a deeper understanding of human language at Marr's algorthmic and implementational levels seems very relevant to the explanatory goals of the Minimalist Program.

References

Aho, A. V., & Ullman, J. D. (1972). *The theory of parsing, translation and compiling; volume 1: Parsing*. Englewood Cliffs: Prentice-Hall.

Berwick, R. C., & Chomsky, N. (2016). *Why only us: Language and evolution the MIT press*. Cambridge: MA.

Chomsky, N. (1959). On certain formal properties of grammars. *Information and Control*, *2*(2), 137–167.

Chomsky, N. (1986). *Knowledge of language: Its nature origin and use*. New York: Praeger.

Chomsky, N. (1988). *Language and problems of knowledge: The Managua lectures, volume 16*. Cambridge: The MIT Press.

Chomsky, N. (1995). *The minimalist program the MIT press*. Cambridge: MA.

Elman, J. (1990). Finding structure in time. *Cognitive Science*, *14*, 197–211.

Fitch, W. T. (2014). Toward a computational framework for cognitive biology: Unifying approaches from cognitive neuroscience and comparative cognition. *Physics of Life Reviews*, *11*(3), 329–364.

Goldwater, S., Griffiths, T. L., & Johnson, M. (2009). A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, *112*(1), 21–54.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, *4*(2), 251–257.

Johnson, M. (1989). Parsing as deduction: the use of knowledge of language. *Journal of Psycholinguistic Research*, *18*(1), 105–128.

Johnson, M. (2013). Language acquisition as statistical inference. In Anderson, S.R., Moeschler, J., & Reboul, F. (Eds.) *The language-cognition interface, pages 109–134. Librarie Droz, Geneva*.

Joshi, A. K., Shanker, K. V., & Weir, D. (1991). The convergence of mildly context-sensitive grammar formalisms. In Wasow, T., Sells, P., & Shieber, S. (Eds.) *Foundational issues in natural language processing, pages 31–81. The MIT Press, Cambridge, MA*.

Kuroda, S.-Y. (1964). Classes of languages and linear-bounded automata. *Information and Control*, *7*(2), 207–223.

Marr, D. (1982). *Vision W.H*. New York: Freeman and Company.

Michaelis, J. (2001). Derivational Minimalism is mildly context-sensitive. In Moortgat, M. (Ed.) *Logical aspects of computational linguistics: third international conference, LACL'98, Grenoble, France, December 14–16, 1998 Selected Papers, pages 179–198, Berlin, Heidelberg. Springer Verlag*.

Pereira, F. C., & Warren, D. H. (1983). Parsing as deduction. In *The Proceedings of the 21st annual meeting of the association for computational linguistics, pages 137–144, MIT, Cambridge, MA*.

Perfors, A., Tenenbaum, J. B., & Regier, T. (2011). The learnability of abstract syntactic principles. *Cognition*, *118*(3), 306–338.

Poggio, T. (2012). The levels of understanding framework, revised. *Perception*, *41*(9), 1017–1023.

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

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist networks. *Artificial Intelligence*, *46*, 159–216.

Smolensky, P., Lee, M., He, X., Yih, W., Gao, J., & Deng, L. (2016). Basic reasoning with tensor product representations. arXiv:1601.02745.

Stabler, E. P. (2013). Two models of minimalist, incremental syntactic analysis. *Topics in Cognitive Science*, *5*(3), 611–633.

Steedman, M. (2000). *The syntactic process*. Cambridge: MIT press.

Steedman, M. (2014). Evolutionary basis for human language: Comment on "toward a computational framework for cognitive biology: Unifying approaches from cognitive neuroscience and comparative cognition" by tecumseh fitch. *Physics of Life Reviews*, *11*(3), 382–388.

Vijay-Shanker, K., Weir, D. J., & Joshi, A. K. (1987). Characterizing structural descriptions produced by various grammatical formalisms. In *The Proceedings of the 25th annual meeting of the association for computational linguistics, pages 104–111. The association for computational linguistics*.

Weir, D. J. (1994). Linear iterated pushdowns. *Computational Intelligence*, *10*(4), 431–439.