Draft only. Do not cite without permission.

Final version due to appear in *Mind*

mark.sprevak@ed.ac.uk

Review of *The Language of Thought: A New Philosophical Direction* by Susan Schneider

Mark Sprevak
University of Edinburgh

3 May 2018

SUSAN SCHNEIDER, *The Language of Thought: A New Philosophical Direction*. Cambridge, MA: MIT Press, 2011, 272 pp., $36.00 (hardback). ISBN 978-0-262-01557-8.

1 Introduction

The Language of Thought (LOT) is closely associated with the work Jerry Fodor. Fodor defended the idea in his book, *The Language of Thought* (1975), and continued to do so, with relatively minor revisions, throughout his career. Susan Schneider's book does not aim to be an exegesis or defence of Fodor. Instead, it offers an alternative to Fodor's version of LOT that Schneider says is an improvement and a worthy replacement. Her aim is to overcome three challenges that face Fodor's version of LOT.

According to both Fodor and Schneider, LOT's goal is to explain human thought in naturalistic, mechanical terms. Schneider defines LOT as a package of three claims to achieve this end. First, having a thought involves tokening symbols in your head and combining those symbols into well-formed symbolic expressions according to language-like grammatical and semantic rules. Second, thinking is a computational process over those LOT symbol and symbolic expressions. Third, the semantic value of an LOT symbol is determined by it standing in a naturalistic causal or nomic 'locking' relation to entities in the world.

Schneider says that Fodor's version of LOT faces three challenges:

1. Central reasoning is not computational
2. The notion of an LOT symbol is unclear
3. LOT is unable to handle Frege cases

I will describe the three challenges and Schneider's proposed solutions. As will become clear, I don't entirely agree with her solutions. Especially with respect to her answer to (2), I think that Schneider's version of LOT incurs costs that raise questions about whether it is the right answer to these challenges. But my criticism should not take away from my overall positive impression of the book. This book will undoubtedly set the agenda for future work on LOT. It places the problem of the nature of LOT symbols – often neglected – at the centre of the LOT debate and it shows how solutions to this problem reach out and touch many other aspects of LOT. The quality of scholarship and writing throughout the book is high. Unusually for a philosophy monograph, it is also fun to read.

2 Central reasoning is not computational

Fodor famously argued against LOT as a theory of central reasoning. Fodor defined central reasoning as nondemonstrative reasoning that is sensitive to all (or nearly all) of one's beliefs. Central reasoning was meant to cover mental processes like how we revise our beliefs in light of evidence, how we make inductive inferences, and how we construct plans to achieve our goals. According to Fodor, two barriers stop LOT from being able to explain central reasoning: the globality problem and the relevance problem.[1]

First, the globality problem. Fodor said that certain properties of individual representations – their simplicity, centrality, and conservativeness – are 'global' in the sense that they vary depending on the context of use. They are not intrinsic to the representations of which they are predicated. Sometimes adding a belief will complicate a plan; sometimes it will simplify a plan. A belief's 'simplicity' does not supervene on that belief's intrinsic properties, and therefore it does not supervene on the belief's syntactic properties. Computational processes are sensitive only to syntactic properties. Therefore, Fodor said, reasoning that requires sensitivity to

[1] These are sometimes misleading described as the 'frame problem'. See Shanahan (1997) for a description of the frame problem.

global properties cannot be a computational process, and so falls outside the remit of LOT.

Schneider responds in a chapter co-written with Kirk Ludwig. Schneider and Ludwig point out (correctly) that a computer can be sensitive to more than just the syntax of one representation; it can also be sensitive to syntactic interactions: how a representation's syntax relates to the syntax of other representations and how it relates to the system's general syntactic rules of manipulation. The failure of an individual representation's simplicity to supervene on its syntax does not mean that that representation's simplicity is not computationally accessible. Simplicity may supervene on syntactic interactions between multiple representations. It is worth noting that Fodor (2000) considers this possibility in a view he labels M(CTM). However, he argues that this solution is prevented from working by the relevance problem, shifting attention to the other part of his argument.[2]

The relevance problem stems from the observation that central reasoning has access to a large number of representations: potentially all of the system's beliefs, desires, and thoughts. Any one of these could be relevant to the system's central reasoning, but usually only a few are. Our own central reasoning system tends to focus on just those representations that are relevant to our current goals, plans, and context. But how does it know *which* representations are relevant without doing an exhaustive, and impracticable, search through its entire database of representations? Fodor says we do not know of any computational method that would solve this problem. (We don't know of any non-computational method either, but never mind that). He says that the relevance problem explains why we have failed to produce a computer with artificial general intelligence (AGI). Successful AI systems excel at narrowly defined tasks (like playing Go or detecting your face), but they do not show general intelligence: they do poorly at putting together plans and strategies outside their narrowly specified competences.

Building on work by Shanahan and Baars (2005), Schneider argues that the solution to the relevance problem can be found within Global Workspace Theory.[3] According to Global Workspace Theory (GWT), multiple 'specialist' cognitive processes compete for access to a global cognitive 'workspace'. If granted access, the information a specialist has to offer is 'broadcast' back to the entire set of specialists. Access to the global workspace is partly controlled by 'attention-like' processes. The contents of the global workspace unfold in a largely serial manner over time. Schneider identifies activities in the global workspace with central reasoning, and she argues

[2] See Samuels (2010) for a helpful reconstruction and critical discussion of Fodor's argument here.

[3] Schneider also discusses its neuronal implementation, the Global Neuronal Workspace Theory (Dehaene and Changeux, 2004).

that the relevance problem is solved by the ceaseless parallel work of the specialists.

I am not convinced by this solution. GWT describes a functional architecture – and in the case of its neuronal version, an anatomical architecture – that the brain could use to share and manage information. GWT clearly pertains to *part* of the relevance problem: in order to bring information to bear in central reasoning there must be channels to share and manage information. But, and it is an important but, GWT does not say how traffic along those channels is regulated to guarantee relevance. It does not explain how relevant, and only relevant, information is shepherded into the global workspace. Baars and Shanahan don't attempt to explain this, and neither does more neurally-orientated GWT work. It is not obvious what GWT should say here. The answer to the relevance problem is not to appeal to bottom-up pressure from the specialists (for there is no reason to think that a specialist who shouts loudest contains relevant information); it also is not to appeal to top-down selection by some executive process (for that would introduce the relevance problem for the executive process). How then does the reasoning system ensure that relevant information, and not irrelevant information, filters into the global workspace? If the answer is 'attention', what mechanism keeps attention aligned to what is relevant to the system in the current context? Baars and Franklin (2003) describe an interplay between executive processes, specialists, and attentional modulation that controls access of information to the global workspace. But we should recognise that this is a hand-waving sketch, not a computational solution to the relevance problem. A solution to the relevance problem may be compatible with GWT. But GWT, as it currently stands, is largely silent about how relevance is computed.

What would it take to solve the relevance problem? Fodor hitches our ability to solve the problem to our ability to build an AGI. Schneider says this sets the standards too high. I do not think so. Building a computational system that can engage in nondemonstrative reasoning shows that we know how to solve the relevance problem; that we *really* know how to solve it and not off-load the hard parts to unexplained aspects of the model ('executive processes', 'attention'). Building an artificial simulation capable of non-trivial nondemonstrative inference is the benchmark that the relevance problem can be solved by computation.

Fodor thought we'd never get there. Citing a history of past failure he argued that the prospects for solving the relevance problem computationally are bleak. However, induction over past failures is only valid if the computational techniques explored so far are representative of computational techniques we will discover in the future. Fodor's confidence in this assumption strikes me as unfounded. Schneider overreaches when she says that GWT 'solves' the relevance problem, but her overall strategy in responding to Fodor strikes me as fundamentally correct: to promote opportunities offered by non-traditional, non-serial computational architectures. There are more computational architectures than were dreamt of in Fodor's philosophy (or than we can even imagine today). GWT is one example of such a non-traditional architecture, but there are many others. Recent empirical work explores some of these. Deep Q-networks, which are completely unrelated to GWT, show promising elements of domain-general reasoning. A single deep Q-network can learn to play 49 Atari computer games, often at super-human levels, switching strategy depending on the game it is currently playing (Mnih et al., 2015). Significantly, the network is never told which game it is playing. It works this out for itself from the pattern of pixels it 'perceives'. The network pulls together by itself a set of plans and strategies relevant for playing the game in hand. This isn't AGI, but it's a step in the right direction.

3 What is a LOT symbol?

LOT tries to make thought less mysterious by explaining it in terms of LOT symbols. But what is an LOT symbol? If you look at someone's brain, or at a readout of their neural activity, you don't see anything that looks like a symbol. How then should we understand LOT's talk of symbols in the head? Fodor said little about this; he instead focused on the benefits to cognitive psychology that accrue once one has already posited LOT symbols. Schneider aptly called the question about symbols the 'elephant in the room' for LOT.

If one is puzzled about some entity, X, a common philosophical gambit is to substitute the question of what X is with a question about X's individuation conditions. This is what Schneider does here. Her question about symbols becomes: When are two physical tokens – in particular, two brain states – of the same LOT symbol type? The answers she considers aim to give us necessary and sufficient conditions for individuating brain states into symbol types.

Schneider discards two theories of LOT symbols before proposing her own.

The first theory she discards is a 'semantic' theory of symbols. A semantic theory of symbols says that two physical tokens are of the same symbol type just in case they have the same semantic content. Schneider's objection is that a semantic account would conflict with LOT's aim of giving a reductive, naturalistic theory of semantic content. According to Schneider, LOT is committed to explaining the semantic content of LOT symbols in terms of naturalistic (causal or informational relations) relations between LOT symbols and the world. This reductive project can work only if all the players in the reductive base – including LOT symbols – do not themselves depend on semantic content.

The second theory Schneider rejects is an 'orthographic' theory of symbols. An orthographic theory of symbols says that two physical tokens are of the same symbol type just in case they have the same 'shape'. Ink marks in written English appear to be grouped into symbol types based on their physical shape. Clearly, 'shape' must mean something different for LOT symbols than it does for written English (you don't find neurons shaped like the letter 'a'). Schneider rejects the orthographic theory because it does not tell us how to understand this alternative notion of 'shape' for brains. Without this, such an account would put us no further forward in explaining what an LOT symbol is.

Schneider preferred theory of symbols is a 'computational-role' theory. Her theory says that two physical tokens are of the same symbol type just in case they play the same computational role within the system. Schneider unpacks the latter condition in terms of the tokens' physical interchangeablity without affecting the computation. Two physical tokens play the same computational role if and only if one physical token can be physically replaced with the other without affecting any (actual or possible) computational transitions of the system. An important source of support for her view comes from John Haugeland's analysis of symbol systems like chess:

> Formal tokens are freely interchangeable if and only if they are the same type. Thus it doesn't make any difference which white pawn goes on which white-pawn square; but switching a pawn with a rook or a white pawn with a black one could make a lot of difference. (Haugeland, 1985, p. 52)

Physical tokens are typed by their computational role. Computational roles are the same if and only if physical tokens are exchangeable without affecting the system's computational transitions. Schneider goes on to argue that tokens should be typed by their *total* computational role. This means that any change, no matter how minor, to a system's (actual or possible) computational transitions produced by physically exchanging two of its physical tokens entails that the tokens are not of the same symbol type.

I will not describe the arguments Schneider gives to support her theory. Instead, I wish to flag two potential problems.

The first is that her theory (and Haugeland's) does not appear to work for more complex systems such as electronic PCs. Inside a PC, physical tokens of the same symbol type vary enormously in their physical nature – they are rarely freely interchangeable. Conversely, physical tokens of different symbol types can sometimes be interchanged without affecting the computation. This is because electronic computers, unlike chess sets, keep track of changing physical tokens and adjust their processing accordingly. This strategy is called 'virtualising' the physical hardware.

It occurs at multiple levels inside a PC.[4] As an example, suppose that a physical token of the symbol type 'dog' is tokened on my PC (maybe as part of an email message I am writing). Imagine that this physical token involves electrical activity in my PC's physical RAM locations 132, 2342, and 4562. However, these physical locations are not somehow reserved for 'dog' within my computer. Nanoseconds later, tokening 'dog' may involve electrical activity in completely different RAM locations, say, 32, 42, 234. Now, tokening 'cat' may involve electrical activity in the physical RAM locations 132, 2342, and 4562. The physical memory inside a modern computer is constantly being remapped to optimise my computer's performance. In such a context, using interchangeablity of physical tokens to individuate symbol types would be hopeless. Tokens may play the same total computational role and not be freely physically exchangeable ('dog' now and 'dog' after a memory remap cannot be exchanged without disrupting the computation), and tokens may be freely exchangeable but play different computational roles ('dog' now and 'cat' after a memory remap can be freely exchanged without affecting the computation).

Physical tokens that fall under the same symbol type change during a PC's computation but the PC's *physical* principles of manipulation of its tokens change correspondingly to counterbalance the effect. The PC's *formal* principles for manipulating symbol types (its algorithm) stays constant throughout.[5] It is as if, during a chess match, the physical board and pieces were reshuffled after every move but the physical principles of movement of the pieces were changed correspondingly to accommodate this physical reorganisation (Black's king's castle now moves to entirely different squares and it is represented by a horse-shaped piece, but it can attack, and be attacked by, the same pieces, and the general state of play in the chess game is unaltered). Only a lunatic would do this during a chess match. But physical reorganisation is both adaptive and common in electronic PCs. One might expect brains to use similar virtualising tactics given their benefits in optimising performance with limited computing resources.

To summarise, the first problem is that 'same total computational role' does not equal 'physical interchangeability', at least for computers that use virtualising strategies. The second problem is Schneider's account does not provide sufficiently stable symbols, at least for computers that learn. Schneider foreshadows difficulties when she says that her proposal makes it hard for symbol types to be shared between different computers. You and I do not undergo exactly the same computational

[4] See Hennessy and Patterson (2011), Ch. 5. Virtualisation reaches new heights with cloud-based virtual machines such as those offered by Amazon Web Services. Physical tokens of a single symbol type within a computation can flicker all over the world during the course of a computation, mixing with tokens of other machines, without disrupting the computation, and without any of these tokens being freely interchangeable. It is a miracle that such systems work, but they do.

[5] When I come to the second problem, I consider cases in which the algorithm changes too.

transitions when reasoning about dogs, so we do not have LOT symbols of the same type (maybe you token DOG_1 and I token DOG_2). In a footnote on page 130, Schneider says similar worries apply within a single human over time. Schneider has in mind relatively slow changes in someone's computational roles over a lifetime. The real difficulty comes, not from slow changes, but from short-term learning events.

Traditionally, the algorithms run by electronic PCs do not change: they are fixed either by hardware or by the program the machine is given. But increasingly machines modify their algorithms in light of new information. Machine learning has become big business. Computers that learn, like AlphaGo, modify their (hugely complicated) algorithms in countless ways in response to learning data (either labelled examples of 'good' behaviour or reward/punishment signals). When learning happens, a computer modifies its rules for processing data; its algorithm before and after a learning event are different. This creates a problem for Schneider's account of symbol identity. Schneider indexes symbol identity to a symbol's total computational role. Total computational role is likely to change across learning events. A change, even a small one, to computational role will ramify. Remember that any change no matter how small to a symbol's computational role affects that symbol's identity. Remember also that the computational role of a symbol includes not only the symbol's actual role but also any possible computational transitions that the symbol could enter into. A learning-induced change to computational roles, even a small one which does not affect the *actual* processing of a given symbol, is almost certain to affect some *possible* computational transition that the symbol could enter into – perhaps by affecting the computational roles of other symbols to which the symbol is related via merely possible computational transitions. Unless the system is so regimented as to have *no* possible computational relations between its symbols (and what would be the point of that?), small changes to computational role will percolate throughout the system, changing symbol identities in their wake. The upshot is that Schneider's symbol types will not survive learning.

Brains are learning systems. Indeed, brains never appear to be 'not' learning; even when they are asleep they use self-generated input to update their processing (O'Neill et al., 2010). It seems reasonable to assume that a brain's computational rules are not fixed but are constantly shifting, taking into account new information and trying out new computational strategies. Schneider's account makes LOT symbol types disappear across these shifts. If LOT symbol types are so short lived, it is hard to see how they could be useful for either science or philosophy.

It is worth emphasising that science needs LOT symbols that are stable across learning events. Recent empirical work on LOT proposes that the brain's learning algorithms perform probabilistic inference over LOT expressions (Piantadosi, Tenenbaum and Goodman, 2016; Piantadosi and Jacobs, 2016). For such algorithms to work, it is crucial that the identity of these LOT expression remains fixed across changes to their computational role so that the learner can rationally explore a space of hypotheses. Learning algorithms need to be defined over stable symbol types that do not themselves disappear during learning events. Interestingly, this work on LOT tends to cite Feldman (2012)'s account of symbol identity, which takes a semantic, broadly referential, approach to explain what makes two (noisy, probabilistic) physical states of the brain instantiate LOT symbols of the same type. Schneider herself switches to a semantic method for individuating states when describing computational principles shared between different humans – on her view a *non-computational* way of individuation condition.

4 Concepts and Frege cases

LOT says that concepts are LOT symbol types and that the semantic value of a concept is purely referential. LOT appears to have problem with Frege cases: it is unable to distinguish, at least on semantic grounds, between co-referring concepts. Fodor's solution to this problem is to say that concepts should be individuated by both their semantic properties and their syntactic properties (Fodor, 2008, Ch. 3). CICERO and TULLY have the same semantic content, but they are distinct concepts because they are two different LOT symbol types.

Schneider argues for essentially the same solution, but she interposes her own theory of LOT symbol types (as mentioned above, Fodor lacked a clear theory about this). The result is a theory of concepts very different from what Fodor intended. Fodor called 'pragmatism' the idea that one's concepts depend on one's cognitive or behavioural capacities (recognitional, classificatory, inferential capacities). To have the concept DUCK is to be able to recognise ducks, classify ducks versus non-ducks, and perform inferences about ducks. Fodor thought that pragmatism was a bad idea; he called it 'the defining catastrophe of analytic philosophy of language and philosophy of mind in the last half of the twentieth century' (Fodor, 2005, pp. 73–74). LOT's solution to Frege cases is to individuate concepts by both their syntactic and semantic properties. But according to Schneider's theory of symbols, a concept's syntactic properties depend on its total computational role. This means that a concept's identity depend on its role in thought – including its role in recognition, classification, and inference. Schneider's theory of LOT symbols takes us from Fodor's LOT to concept pragmatism.

There is delicious irony here, but should a fan of LOT accept Schneider's theory of concepts? While not disputing her arguments, I would like to strike a note of caution. Schneider's theory would make an agent's concepts as unstable, idiosyncratic, and ephemeral as her LOT symbol types. Schneider says shared referents provide semantic stability. But this is cold comfort in this context. An agent needs stable concepts in order to do valid inference. The same concepts need to be tokened in the agent's premises and her conclusions in order for her inference to be valid. The risk is that this won't happen, or at least it won't happen often on Schneider's view. The concepts that are tokened in the premises most likely won't be around by the time the agent tokens her conclusion. If an agent were to learn just one thing between tokening premises and conclusion, then her inference would likely be invalid as her concepts would have changed. The purpose of LOT is to mechanise rational thought and concepts need to be stable for this. They need to hang around long enough for agents to use them repeatedly. Individuating concepts by their total computational role does not provide stable enough foundations for LOT to achieve its goal.

5 Conclusion

This book throws into relief just how hard, and important, is the question of individuating LOT symbols. My own inclination is to give a semantic answer to this question. Unlike Schneider, I'm not worried about presupposing semantic content in an account of symbols as I think that reductive, naturalistic accounts of semantics already face more serious problems than a semantically-inflected notion of symbols. In any case, LOT should be decoupled from the project of coming up with a reductive theory of semantic content. LOT may be true and useful independent of the fate of this naturalisation project. Indeed, cognitive scientists who use LOT do not care much for this naturalising project at all.

Schneider's book advances the debate on LOT. She updates LOT by integrating diverse considerations ranging from neurocomputational models to neo-Russellianism about names. She wears her learning lightly, engaging the reader with simple examples and clearly motivated considerations. Whether you end up agreeing with all her claims or not, I would encourage you to buy and read this book.

Bibliography

Baars, B. and S. Franklin (2003). "How conscious experience and working memory interact". In: *Trends in Cognitive Sciences* 7, pp. 166–172.

Dehaene, S. and J.-P. Changeux (2004). "Neural mechanisms for access to consciousness". In: *The Cognitive Neurosciences, III*. Ed. by M. Gazzaniga. Cambridge, MA: MIT Press, pp. 1145–1157.

Feldman, J. (2012). "Symbolic representation of probabilistic worlds". In: *Cognition* 123, pp. 61–83.

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

— (2000). *The Mind Doesn't Work That Way*. Cambridge, MA: MIT Press.

— (2005). *Hume Variations*. Oxford: Oxford University Press.

— (2008). *LOT2: The Language of Thought Revisited*. Oxford: Oxford University Press.

Haugeland, J. (1985). *Artificial Intelligence: The Very Idea*. Cambridge, MA: MIT Press.

Hennessy, J. L. and D. A. Patterson (2011). *Computer Organization and Design: The Hardware/Software Interface*. 4th ed. Waltham, MA: Morgan Kaufmann.

Mnih, V., K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al. (2015). "Human-level control through deep reinforcement learning". In: *Nature* 518, pp. 529–533.

O'Neill, J., B. Playdell-Bouverie, D. Dupret and J. Csicsvari (2010). "Play it again: Reactivation of waking experience and memory". In: *Trends in Neurosciences* 33.220–229.

Piantadosi, S. T. and R. A. Jacobs (2016). "Four problems solved by the probabilistic language of thought". In: *Current Directions in Psychological Science* 25, pp. 54–59.

Piantadosi, S. T., J. B. Tenenbaum and N. D. Goodman (2016). "The Logical Primitives of Thought: Empirical Foundations for Compositional Cognitive Models". In: *Psychological Review* 123, pp. 392–424.

Samuels, R. (2010). "Classical computationalism and the many problems of cognitive relevance". In: *Studies in History and Philosophy of Science* 41, pp. 280–293.

Shanahan, M. (1997). *Solving the Frame Problem*. Cambridge, MA: Bradford Books, MIT Press.

Shanahan, M. and B. Baars (2005). "Applying global workspace theory to the frame problem". In: *Cognition* 98, pp. 157–176.