

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

Mark Sprevak
University of Edinburgh

14 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 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 LOT symbols 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 a computation
2. The notion of an LOT symbol is unclear
3. LOT is unable to handle Frege cases

In this review, I will describe the three challenges and Schneider's proposed solution. As will become clear, I don't entirely agree with everything Schneider says. Especially with respect to her answer to (2), I think that her version of LOT incurs costs that should lead us to question its truth. But my criticism should not take away from my overall positive impression of the book and the project. This book will undoubtedly set the agenda for future work on LOT. It places the often ignored problem of the nature of LOT symbols 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 a computation

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 is meant to cover processes such as how we revise our beliefs in light of evidence, how we make inductive inferences, and how we construct practical plans to achieve our goals. According to Fodor, two problems 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 these properties vary with the context of use; they are not intrinsic to the representations of which they are predicated. Sometimes adding a certain belief to one's belief set will complicate a plan. Sometimes it will simplify a plan. A belief's 'simplicity' does not supervene on that belief's intrinsic properties. Therefore, it does not supervene on that belief's syntactic properties. Computational processes are sensitive only to syntactic properties. So, says Fodor, reasoning that requires

¹These are sometimes misleadingly called the 'frame problem'. See Shanahan (1997) for a description of the frame problem.

sensitivity to 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, that a computer is not just sensitive to the syntax of individual representations. A computer is also sensitive to syntactic relations: how a representation's syntax relates to the syntax of other representations and how these relate to the system's general rules of syntactic manipulation. The failure of an individual representation's simplicity to supervene on its syntax does not mean that the representation's simplicity cannot be tracked by a computational process. Simplicity may supervene on (and be computationally tracked by following) syntactic interactions between representations. It is worth noting that Fodor (2000) considers this possibility too in a view he labels M(CTM). However, Fodor argues that this solution would then run into the relevance problem, shifting attention to the other part of his argument.²

The relevance problem arises because central reasoning has access to a large number of representations: all of the system's beliefs, desires, and thoughts. Any one of these could be relevant to the system's reasoning about any given case, but usually only a few are. The human central reasoning system tends to focus on just those representations that are relevant to the agent's current goals, plans, and context. But how does it know *which* representations are relevant without doing an exhaustive, impracticable search through its entire database? 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 our failure to produce a computer with artificial general intelligence (AGI). Successful AI systems tend to excel at narrowly defined tasks (like playing Go or detecting your face), but they do not show general intelligence: they are poor at pulling together relevant information to make plans outside their narrowly defined area of competence.

Building on work by Shanahan and Baars (2005), Schneider argues that a solution to the relevance problem can be found within Global Workspace Theory.³ The Global Workspace Theory (GWT) says that 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 set of specialists. Access to the global workspace is 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 that the relevance problem is solved by the ceaseless parallel work of the specialists.

²See Samuels (2010) for a helpful reconstruction and criticism of Fodor's argument here.

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

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 pertains to *part* of the relevance problem: in order to bring information to bear in central reasoning there must be channels to share 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 typically 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. The answer is not 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 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 only relevant 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 interplay between 'executive functions', 'specialist networks', and 'attention codelets' that controls access of relevant information to the global workspace. How these components work to track relevance is left largely unspecified. As an answer to the relevance problem, this seems more like hand waving than a computational solution. A computational 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 find a computational solution to the relevance problem? Fodor pegs our ability to do this to our ability to build an AGI. Schneider says that this sets the bar too high. I do not think so. Building a computational model 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 an unexplained part of the model ('executive function', 'attention'). Building a computational simulation capable of nondemonstrative inference is the hallmark that the relevance problem can be solved.

Fodor thought we'll never get there and he cited a long history of past computational failures. However, past failures are only a guide to the future if the computational techniques explored so far are representative of computational techniques we will discover in the future. Fodor's confidence in this strikes me as unfounded. Schneider may overreach when she says that GWT solves the relevance problem, but her overall strategy – promoting the opportunities offered by novel computational architectures – strikes me as fundamentally correct. There are more computational architectures than were dreamt of in Fodor's philosophy (or than we can dream of today). GWT is one example, but there are many others. Deep Q-networks, completely unrelated to GWT, shows promising elements of domain-general reasoning. A single

deep Q-network can 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 ‘sees’. The network pulls together, by itself, a plan and strategy relevant for playing the game in hand. This isn’t AGI or a solution to the relevance problem, but it’s a step in the right direction.

3 What is a LOT symbol?

LOT explains thought and thinking in terms of LOT symbols. But what is an LOT symbol? If you look inside someone’s brain you don’t see anything that looks like a symbol. How then should we understand LOT’s talk of symbols in the head? Schneider calls this question the ‘elephant in the room’ for LOT. Fodor said little to answer it. He focused instead on arguing for explanatory and predictive benefits which he thought would accrue once one posits brain-based LOT symbols, whatever they turned out to be.

If one is puzzled about some thing, a common gambit is to substitute the question of what that thing is with a question about its individuation conditions. This is what Schneider does here. Her question becomes: When are two physical tokens – two brain states – of the same LOT symbol type?

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

The first theory she discards is a ‘semantic’ theory. A semantic theory of LOT symbols says that two physical tokens are of the same symbol type just in case they have the same semantic content. Schneider’s objection to this is that a semantic theory of symbols would conflict with LOT’s ambition to give a reductive, naturalistic theory of semantic content. 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 won’t work if one of the players in the reductive base – LOT symbols – themselves depend on semantic content.

The second theory Schneider rejects is an ‘orthographic’ theory. An orthographic theory says that two physical tokens are of the same symbol type just in case they have the same ‘shape’. The ink marks on this page can be grouped into symbol types based on their physical shape. ‘Shape’ clearly means something different for LOT symbols inside the brain – you don’t find neurons shaped like the letter ‘a’. Schneider rejects the orthographic theory because it does not provide any information about this alternative notion of ‘shape’ for brain-based symbols. Absent this, an orthographic

account is largely empty as a theory of LOT symbol types.

Schneider preferred theory is a ‘computational role’ theory of LOT symbols. This 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 this second condition in terms of the tokens being physical interchangeable without affecting the computation. Two physical tokens play the same computational role just in case one physical token can be swapped with the other without affecting any (actual or possible) computational transitions of the system. A key source of support for her view comes from John Haugeland’s account 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. These computational roles are the same if and only if physical tokens could be swapped without affecting the system’s actual and possible computational transitions. Schneider argues furthermore that tokens should be typed into symbols by their *total* computational role. That means that any change, no matter how small, to a system’s (actual or possible) computational transitions resulting from an exchange of two of its physical tokens entails that those 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 computers like modern electronic PCs. Inside a PC, physical tokens of the same symbol type vary enormously in their physical nature and they are rarely freely interchangeable. Conversely, physical tokens of different symbol types can sometimes be interchanged without affecting the computation at all. This is because modern PCs, unlike chess sets, keep track of changes in their physical tokens and adjust their physical processing accordingly. This strategy is called ‘virtualising’ the physical hardware. It occurs at multiple levels inside a PC.⁴ For example, suppose that a physical token of the symbol type ‘dog’ is tokened on my PC (maybe as part of an email message). Imagine that this physical token involves electrical activity in my PC’s physical RAM locations 132, 2342, and 4562. But this pattern of physical states is not somehow reserved for ‘dog’ tokens within my computer. Nanoseconds later, tokening ‘dog’ may involve electrical activity in different physical RAM locations,

⁴See Hennessy and Patterson (2011), Ch. 5. Virtualisation reaches its height with cloud-based computers such as those offered by Amazon Web Services.

say, 32, 42, 234. Tokening ‘cat’ may now involve electrical activity in the old physical RAM locations of 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 interchangeability of physical tokens over the computation to individuate symbol types would be hopeless. Tokens that play the same total computational role may not be freely physically exchangeable (‘dog’ now and ‘dog’ after a memory remap), and tokens that are freely exchangeable without affecting the computation may play different computational roles (‘dog’ now and ‘cat’ after a memory remap).

While a modern PC computes, physical tokens that fall under the same symbol type vary but the PC’s *physical* principles of manipulation vary accordingly to counterbalance the effect. The PC’s *formal* principles for manipulating symbol types (its algorithm) stays constant throughout.⁵ Imagine that during a chess match the physical board and pieces were reorganised after every move but the principles of movement of the physical pieces were changed correspondingly to accommodate the reorganisation: Black’s king’s castle can now move to different squares and it is symbolised by a horse-shaped piece, but it can attack, and be attacked by, the same opposing pieces, and the state of play in the match is unaltered. Only a lunatic would do this physical remapping during a chess match. But physical remapping is both adaptive and common in electronic PCs. One might expect brains to use similar virtualising tactics given their proven benefit in optimising performance with limited computing resources.

In sum, the first problem is that ‘same total computational role’ does not mean ‘physical interchangeability’, at least for computers that use virtualising strategies. The second problem is Schneider’s account does not provide stable symbol types. Schneider foreshadows difficulties here when she says that her proposal makes it hard for symbol types to be shared between different computers. You and I are not disposed to undergo exactly the same computational transitions when thinking about dogs, so we do not have LOT symbols of the same type (maybe you have DOG₁ and I have DOG₂). In a footnote on page 130, Schneider says similar worries apply inside a single human being over time. Schneider has in mind relatively slow changes in computational roles that occur over someone’s lifetime. However, the difficulty comes, not from these slow changes, but from short-term changes produced by learning.

The algorithms run by electronic PCs are normally fixed, either by their hardware or by the program they are given. But machines can also modify their algorithms. Machine learning has become big business. Computers like AlphaGo modify their (hugely complicated) algorithms in many ways in response to learning data (either

⁵Later, I consider cases in which the algorithm changes too.

labelled examples of ‘good’ behaviour or reward/punishment signals). When learning occurs, a computer modifies its algorithm; total computational roles before and after learning 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 changes across learning events. A change, even a small one, to a symbol’s computational role will ramify. Remember that any change, no matter how small, affects a symbol’s identity. Remember too that the computational role of a symbol includes not just the symbol’s actual computational transitions but also any possible computational transitions that it could undergo. A change induced by learning, even a small one which does not affect the actual processing of a given symbol type, is almost certain to affect some *possible* transition that the symbol type could enter into – even if only by affecting the computational roles of other symbols to which the symbol is related by possible computational transitions. Unless the system is so designed as to minimise all the computational relations between its symbol types (and what would be the point of a computer like that?), small changes to computational role will percolate throughout the computational system, changing symbol identities in their wake. The upshot is that Schneider’s symbol types are unlikely to survive learning.

Brains are learning computers. Indeed, brains never appear to be ‘not’ learning – they appear to keep learning even when they are asleep (O’Neill et al., 2010). It seems reasonable to assume that a brain’s computational roles are not fixed but are constantly shifting, adapting to 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 ephemeral, it is hard to see how they could be useful to science or philosophy.

Science needs LOT symbols that are stable across learning. Recent 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). In order for such algorithms to work, it is essential that the identity of LOT expressions remains fixed across changes to their computational role so that the learner can explore a consistent space of hypotheses. Learning algorithms need to be defined over stable symbol types that do not themselves change during learning. Interestingly, this work tends to cite Feldman (2012)’s account of LOT symbol identity, which takes a semantic, broadly referential, approach to explain what makes two (noisy, probabilistic) brain states of the same LOT symbol type. Schneider herself switches to a semantic way of individuating brain states when describing computational principles shared between different humans.

4 Concepts and Frege cases

According to LOT, concepts are LOT symbols and the semantic value of a concept is purely referential. LOT appears to have a *prima facie* problem with Frege cases: it cannot distinguish between co-referring concepts, at least on purely semantic grounds. Fodor's solution to this was 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 value, but they are distinct concepts because they are two different LOT symbol types.

Schneider argues for essentially the same solution as Fodor, but she inserts her own theory of LOT symbol types. The result is a theory of concepts very different from what Fodor intended. Fodor called 'pragmatism' the claim that one's concepts depend on one's cognitive or behavioural capacities (recognitional, classificatory, inferential capacities). For a pragmatist, 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 '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). According to Schneider's theory of LOT symbols, a concept's identity depends on its total computational role. This means that a concept's identity depends on its role in many aspects of our thought – including its role in recognition, classification, and inference. The upshot is Schneider's theory of LOT symbols commits LOT to concept pragmatism.

There is delicious irony here, but should we 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 just as ephemeral as her LOT symbol types. Schneider says that unchanging semantic (referential) content provides stability. But an agent needs stable concepts, not just stable referents, in order to do valid inference. The same concepts need to appear in an agent's premises and her conclusions in order for her inference to be valid. This won't happen, or at least it won't happen often, on Schneider's view. The concepts that are tokened in the premises probably won't exist by the time the agent gets around to tokening her conclusion. If an agent were to learn just one new thing between tokening the premises and tokening the conclusion, then her inference would be invalid as her concepts would have changed. The purpose of LOT is to mechanise thought and concepts need to be stable for this. They need to hang around long enough for agents to use them multiple times. Individuating concepts by their total computational role does not provide stable enough concepts 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. In contrast to Schneider, my inclination is to give a semantic answer to this question. I'm not too worried about presupposing semantic content in an account of symbols as I think that reductive accounts of semantics already face more serious problems than a semantically-inflected notion of symbols. In any case, it makes sense for LOT to be decoupled from the project of finding a reductive naturalistic theory of content. LOT may be true and useful independent 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 considerations as diverse as neurocomputational models and neo-Russellianism about names. The book wears its learning lightly, engaging the reader with simple examples and clearly motivated considerations. Whether you end up agreeing with all its 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: Harvard 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.