Philosophical Psychology
Vol. 19, No. 6, December 2006, pp. 795–821

Routledge
Taylor & Francis Group

Two Concepts of "Form" and the So-Called Computational Theory of Mind

John-Michael Kuczynski

According to the computational theory of mind (CTM), to think is to compute. But what is meant by the word 'compute'? The generally given answer is this: Every case of computing is a case of manipulating symbols, but not vice versa—a manipulation of symbols must be driven exclusively by the formal properties of those symbols if it is qualify as a computation. In this paper, I will present the following argument. Words like 'form' and 'formal' are ambiguous, as they can refer to form in either the syntactic or the morphological sense. CTM fails on each disambiguation, and the arguments for CTM immediately cease to be compelling once we register that ambiguity. The terms 'mechanical' and 'automatic' are comparably ambiguous. Once these ambiguities are exposed, it turns out that there is no possibility of mechanizing thought, even if we confine ourselves to domains (such as first-order sentential logic) where all problems can be settled through decision-procedures. The impossibility of mechanizing thought thus has nothing to do with recherché mathematical theorems, such as those proven by Gödel and Rosser. A related point is that CTM involves, and is guilty of reinforcing, a misunderstanding of the concept of an algorithm.

Keywords: Computational Theory Of Mind; Formal Operation; Jerry Fodor; Syntax; Token-Semantics

1. Introduction

There is good reason to believe that mental entities are identical with, or realized by, neuro-cerebral entities. If this is right, then your being in pain or thinking about the number two is realized by some brain event or series of such events (Fodor, 1981, pp. 2–3).[1]

Correspondence to: John-Michael Kuczynski, Department of Philosophy, 5631 South Hall, Santa Barbara, CA 93106, USA. Email: jsbach@jps.net

ISSN 0951-5089 (print)/ISSN 1465-394X (online)/06/060795-27
DOI: 10.1080/09515080601001887

Brains are obviously governed by the laws of physics—just like billiard balls and gases. At the same time, our thought processes are obviously *rational*, at least up to a point. A person who believes that Bill is either in Greenland or Iceland, and that he is not in Iceland is likely to form the belief that Bill is in Greenland. To be sure, we are not *always* rational. But thoughts and, more generally, mental contents succeed one another in ways that are, to a high degree, consistent with the demands of rationality (Fodor, 1981, pp. 11–16, 1987, pp. 18–19, 1994, p. 294).

There is a problem. The laws of physics are purely descriptive, not prescriptive. In moving this way as opposed to that, the billiard ball isn't doing anything right or wrong, it isn't being rational or irrational. Physics alone decides what the billiard ball does. Consciousness of norms—or of canons of good reasoning—plays no part. Our brains are governed by the laws of physics no less than billiard balls. The laws of physics alone decide how they will behave. There is no room for awareness of the canons of good reasoning to play a part. And our mental lives are entirely realized by our brain activity.[2] But our mental activity obviously *conforms* to such canons, at least to a high degree.[3]

So we are left with a problem. Our mental activity is rational, i.e., it proceeds in ways that are countenanced by canons of good reasoning. At the same time, our brains, and therefore our minds, are governed entirely by the laws of physics. The laws of physics are not laws of rationality. So laws of rationality play no part in our thinking. There is an elegant solution to this problem. This solution constitutes one of the more compelling arguments for the *computational theory of mind* (CTM). This solution is most clearly and forcefully articulated by Jerry Fodor.

The purpose of this paper is to show that this solution involves a massive confusion, and is therefore no solution at all. Thus at least *one* of the arguments for CTM, or at least for one version of it, proves to be quite fallacious. I leave it open to what extent this fact generalizes; I leave it open whether it nullifies CTM or merely removes one of its supporting pillars.[4]

2. Fodor's Solution

From one viewpoint, a symbol is a purely physical entity: a noise, a deposit of ink, a pattern of light on a monitor. Being physical entities, symbols (or symbol-tokens, strictly speaking) obey the laws of physics; their interactions are entirely physics-driven. To explain those interactions, it isn't necessary to invoke anything other than physical entities and physical laws.

But it is easy to create an environment in which the interactions among symbols "track" the laws of logic. Such environments are created and exploited everyday. They are called "computers" (Fodor, 1981, Introduction, especially p. 22; 1987, ch. 1, especially pp. 18–20, 1990; Pylyshyn, 1984).

Consider the symbols '*P*', '*Q*', and 'if *P* then *Q*'. Those symbols have various physical properties. Because they have those properties, the laws of physics dictate that they act in certain ways in certain environments. We can construct an environment in which those symbols interact in the *right* way: an environment in which '*P*' and 'if *P* then *Q*' is followed by '*Q*'. We can construct environments in which symbols follow on another in a way consistent with the demands of rationality. Such environments are entirely physical; and the behavior of the symbols in question is entirely physics-driven. But the structure of the environment guarantees that those entirely physics-driven processes conform to the demands of logic. If the input is '(*P* or *Q*) and not *Q*', the output will be '*P*'. But the process is entirely physics driven: through clever engineering we have ensured that, given the physics involved, output will be logically appropriate to input.

We have thus dissolved the paradox earlier described. There doesn't seem to be any difficulty creating an environment in which entirely physics-driven processes respect the canons of logic—in which the descriptive and the prescriptive coalesce.

But there is an obvious response:

> It is easy enough to create a purely physical environment which *mimics* or *simulates* thought: an environment in which '*P*' and 'if *P* then *Q*' are followed by '*Q*'. But that is not enough for the existence of thought or inference. To infer '*Q*' from '*P*' and 'if *P* then *Q*' is to recognize that the latter two symbols stand in a certain logical relation to the first. The essence of inference is *recognition* of some bearing relation: a relation of entailment or probabilification. Let us tell a story. You believe *P*; you can also believe (if *P* then *Q*); and these beliefs *cause* you to believe *Q*. Have you made an inference? Not necessarily. It is not enough that the first two beliefs bring about the third: they must bring it about in the right way: they must do so by way of your recognizing the existence of some bearing relation. Now let us tell a different story. You believe *P* and also (if *P* then *Q*). You also recognize that *Q* is entailed by *P*, conjoined with (if *P* then *Q*). These beliefs conspire with this recognition to make you believe *Q*. *Here* we have an inference. To have a rational thought-process, it is not enough for symbols to follow one another in a certain order. It is also necessary that their falling into that order embody an awareness of bearing-relations holding among them. In a computer, the symbols follow one another in the right order; but their doing so doesn't involve a recognition of any bearing relations holding among those symbols. So the computer doesn't think at all.

Fodor has a clever response to this kind of objection. What is a computation? It is an inference: but not just *any* kind of inference. If you infer 'Fred is an animal' from 'Fred is an elephant', you have made an inference; indeed, you have made a correct inference. But you have not performed a computation. Why not? Because the inference in question was not purely *formal*; it involved consideration of the *semantics*, and not just the *syntax*, of the symbols involved. A computation is a purely *formal* inference; it is an entirely syntax-driven inference (Fodor, 1987, ch. 1, 1994, p. 86). Consideration of semantics can play no part.

From a purely formal or syntactical viewpoint,

1. 'Fred is an elephant, therefore Fred is an animal'
   is indistinguishable from
2. 'Fred is an elephant, therefore Fred is a smoker'

Of course, (1) is a valid inference, whereas (2) is not. But to see this one must know the *meanings* of the symbols involved. So (1), though valid, is not a formal inference. A purely syntax-driven transition from 'Fred is an elephant' to 'Fred is an animal' is not valid. Thus, it is no computation.

Matters are different with:

3. 'Fred is an elephant; therefore Fred is either an elephant or an aardvark'

To know that (3) is true, you don't have to know what is meant by 'elephant' or 'aardvark' or 'Fred'. (3) is formally true. A purely syntax-driven transition from 'Fred is an elephant' to 'Fred is either an elephant or an aardvark' constitutes a valid inference. It is a computation.

The syntax of an expression is its *form*. Ultimately, form reduces to morphology.[5] Morphology is shape. Shape is entirely physical. A morphology-driven process is an entirely physics-driven process. But some morphology-driven processes are also *inferences*: they are *formal* inferences; they are computations. So entirely physics-driven processes can qualify as inferences: they can qualify as things which can meaningfully and even correctly be described in normative terms—as right or wrong, justified or unjustified.

So it is easy to dissolve the paradox mentioned above. We advocate CTM: we assume that brains (or brains-cum-bodies) are environments that house morphology-driven interactions among symbols. Such interactions are purely physics-driven, but they also qualify as inferences of a certain kind: formal inferences—inferences that embody sensitivity to syntactic, and thus to logical, form. Because of the tight relation between an expression's logical form and its syntax, and between its syntax and its morphology, it follows that entirely morphology-driven operations can constitute inferences, and thus cognitive achievements, of a certain kind. So CTM dissolves the paradox mentioned a while ago (see Fodor, 1981, Introduction, 1987, ch. 2, 1990, 2001, pp. 3–45).[6]

3. The Purpose of this Paper

I wish to show that the line of thought just presented is entirely spurious. There are many problems with it. One problem is that it involves a confusion of two entirely different meanings of the word 'form'. There is form in the syntactic sense, and there is form in the *physical* sense. These two things have practically nothing to do with each other. Any connections between them are exceedingly indirect. Two physical entities that have exactly the same syntactic form can have nothing in common at the level of morphology. And two morphologically identical physical entities can have nothing in common at the level of syntax. So morphology-driven operations, though "formal" in one sense, are by no means syntax-driven. Genuinely syntax-driven operations can *involve* sensitivity to morphology (in the sense of physical shape[7]); but, as we will see, sensitivity to morphology plays a decidedly subordinate role in genuinely syntax-driven operations. And it is, at best, extremely misleading to say that syntax-driven operations can be identified, even to a first approximation, with morphology-driven ones.

Further, a syntax-driven process *is* a semantics-driven process. Any inference from 'Fred is an elephant' to 'Fred is an elephant or Fred is an aardvark' is replete with knowledge of semantics—hardly less so than the inference from 'Fred is an elephant' to 'Fred is an animal'. A syntax-driven operation can no more be identified with a morphology-driven operation than can a semantics-driven operation.

Syntax, we will see, is an aspect of semantics. The syntax of a sentence (or sentence-token) is identical with the more generic facts about its semantics.[8] Semantics consists in word-world relations. For a being to engage in syntax-driven operations, it must have semantic knowledge; and this involves having extra-linguistic knowledge. The ability to engage in syntax-driven operations thus presupposes a wealth of knowledge and, therefore, of cognitive and representational abilities. So the ability to engage in syntax-driven operations cannot possibly have a cognitively foundational role.[9]

There is another, altogether different problem to consider. Syntactic form does *not* typically track logical form, contrary to what used to be thought. Recent work in logic and semantics has demonstrated this (Kaplan, 1979, 1989; see also Barwise & Perry, 1999). Nearly all sentences have a context-sensitive component; and this fact creates a chasm between syntactic and logical form, and it makes it impossible for both semantics and also logical form to be adequately mirrored by syntax. This, in turn, makes semantic, and extra-semantic, knowledge a pre-requisite for being able to engage in operations that track, or embody any sensitivity to, logical form.

For these reasons, the idea that computing could play a foundational role in thought is simply confused. In fact, computing is an extremely special and derivative form of thought. Granted, there is *a* sense in which a being lacking semantic, or extra-semantic, knowledge can "compute." But that sense of the word 'compute' is entirely irrelevant to attempts to explain the foundations of thought. Failure to distinguish real computing, which presupposes a wealth of cognitive abilities, from faux-computing, which does not, is abetted by a tendency to wrongly identify form in the syntactic sense with form in the morphological sense.

4. Syntax Versus Semantics

Consider the sentence:

4. 'If Fred is an elephant, then Fred is either an elephant or Fred is a chipmunk'

Supposedly, this sentence is *formally* true. But what does that mean? Does it mean that its truth has nothing to do with its semantics? That would be nonsense.

Let us start with the obvious. The symbol 'or' (or 'either . . . or . . .') could mean anything; it could mean *and* (or *both . . . and . . .*); it could mean *because*; it could refer to Plato; or it could mean nothing at all.

To know that (4) is true, one must know the semantics of 'or'; one must know what it means. One must know that it doesn't refer to Plato or denote the relation of cause and effect. So to recognize that (4) is "formally" true, one must have knowledge of semantics.

Some expressions are said to be syntax-defining, while others are not. Sometimes the former (or, at least, a certain subset of the former) are referred to as "logical particles." These include 'and', 'if . . . then . . .', 'not'. Given a sentence containing one of these terms, if you replace that term with some other term, you have changed the syntactic structure of that sentence. Consider (3). Suppose you replace the 'or' with an 'and' or a 'because'. In that case, you have changed the syntactic structure of the sentence. Further, you will have turned a sentence that is logically true into one that is not logically true—into one that is, in fact, logically false.[10]

Expressions like 'Fred' and 'elephant' are said *not* to be syntax-defining. Suppose that, in (4), you (uniformly) replace 'Fred' with 'Jerry'. In so doing, you have *not* changed the syntax of the sentence: and the sentence you produce will be logically true exactly if the original sentence had that property. Apparently, the property of logical truth is preserved under (uniform) substitutions of nonlogical expressions for nonlogical expressions.

Nonlogical expressions like 'Fred', 'Jerry', and 'elephant' denote individuals or properties. Logical particles (e.g., 'or', 'for some') are typically thought *not* to denote anything; they are said to have a purely "formal" or "syntactic" function. There are two points to make in response to this. First, even if they do not denote anything, one still has to know their semantics to know that sentences like (4) are logically (or formally) true. You could be the most brilliant logician in the world. If you don't know what is meant by 'or' or 'therefore', you simply won't have any way of knowing that (4) is logically true. And there is no way to read the semantics of such expressions off of their morphology. An expression with the same semantics as 'or' could have any morphology at all. Let *L* be a language just like English except that the *L*-word for 'or' is 'and' and the *L*-word for 'and' is 'or'. In *L*, (4) is logically *false*, not logically true. Let *L** be a language just like English except that, in *L**, 'or' means 'banana'. In *L**, (4) is meaningless; it isn't true or false. So even if syntax-defining terms are nonreferential, one must still know a great deal about semantics to determine if a sentence is logically true or not.

There is another point to make. A case can be made that syntax-defining terms *do* refer. I myself have no doubt that this is so; and it is the view of many logicians and linguists. Such terms can be seen as referring to second-level functions or properties. Consider the sentence:

5. 'Sally is a square or Sally is not a square'

This is logically, and perhaps semantically, identical with:

6. Or*<Sally is a square, Sally is not a square>

where 'or*' denotes a function that assigns truth a sequence of sentences exactly if at least one of them is true. So 'or*' can be seen as denoting the property of being a sequence of sentences such that at least member of that sequence is true. (6) is obviously logically equivalent with (5). So the 'or' in (5) can be seen as a variant of 'or*'; it can be seen as denoting the second-level function (or property) just described.[11]

Exactly similar remarks apply to all syntax-defining terms. Consider the sentence:

7. 'Snow is not green'

This is equivalent to

8. Not*<snow is green>

where 'not*' denotes a function that assigns truth to a sequence exactly if that sequence comprises one sentence and that sentence is false. By reasoning similar to that given a moment ago, the 'not' in (7) can be seen as denoting that second-level function. On similar grounds, Frege (1892a, 1892b) argued that quantifiers denote second-level functions. ('For some x' denotes a function that assigns truth to an open sentence exactly if some object satisfies it; 'for all x' denotes a function that assigns truth to an open sentence or propositional function exactly if nothing fails to satisfy it.) Richard Montague (1974) generalized Frege's analysis to other connectives (see also McCawley, 1981). Many contemporary linguists and philosophers of language (myself included) see tense- and case-markers as denoting second-level functions. Despite years of tradition to the contrary, the distinction between categorematic and syncategorematic items—lexical and nonlexical items—is plausibly seen as a distinction *within* the class of referring terms. 'Or', 'not', and the like, denote second-level functions; 'Fred', 'rabbit', and the like denote individuals or first-order functions.

To sum up, a case can be made that so-called syntactical expressions *do* refer. If this is correct (and, as we saw, even if it isn't correct), then one plainly needs knowledge of semantics to engage in a purely syntax-driven piece of reasoning.

5. Syntactic Form and Nonsyntactic Terms

But let us leave aside consideration of terms like 'or' and 'not'. Even with this waiver, one must *still* know an enormous amount about the semantics of (4) to know that it expresses a logical truth. Consider a language *L* that is just like English except that, in *L*, 'Fred' has the meaning borne by the English word 'and'. In *L*, (4) is simply meaningless; for it has the same meaning as the English pseudo-sentence:

9. 'If and is an elephant, then and is either an elephant or and is not elephant'

Thus (4) expresses a logical truth only because 'Fred' has a quite specific kind of meaning, only because it falls into the semantic category of singular term. Here is another way of looking at it. Let *L** be a language just like English except that, in *L**, 'Fred' has the meaning borne by the English term 'rabbit'. In *L**, (4) is meaningless; for it has the same meaning as the English sentence:

10. 'If rabbit is an elephant, then rabbit is either an elephant or rabbit is not elephant'

(4) expresses a logical truth in English only because 'Fred' has an extremely specific *kind* of meaning: it denotes an individual. If it denoted a second-order function ('and') or a sortal ('rabbit'), then (4) wouldn't be true or false, let alone logically true; it would be ill-formed and meaningless.

The so-called "logical particles" in (4) are not the only ones that are responsible for its syntactic profile. 'Fred' denotes an individual. If it didn't, then (4) would be ill-formed, and would thus have an entirely different syntactic profile from the one it actually has. To recognize that (4) is logically true, one must know that 'Fred' is a certain kind of noun-phrase of some kind; actually, one must know that it is an extremely specific kind of noun-phrase.

Granted, one doesn't have to know *which* exact individual 'Fred' refers to; one need only know that both occurrences of 'Fred' refer to *somebody* or other. (Actually, one must also know that they *corefer*: a fact that, as we will see, deepens the connection between semantics and logical form and that further scuttles the attempt to make syntax-driven operations have a foundational cognitive role.) But one must know that 'Fred' falls into a highly restrictive semantic category; and there is no way to read this fact off of the morphology of that expression.

A physical object—a deposit of ink, a series of hand signals—may have one meaning in English, a different meaning in Albanian, and no meaning at all in Greek. Consider the following sentence-token:

11. 'Snow is white'

In English, that token means *snow is white*. Let *L* be a language in which (11) means: *if Adam Smith's economic theories were correct, then the democratization of South America would not have been nearly as tortured a process as it actually was.* Considered as an expression of *L*, (11) has one syntactic structure; considered as an expression of English, it has a radically different syntactic structure. Given any physical object, no matter what its morphology, it could have any syntactic structure at all. Indeed, given any physical object *O*, no matter what its morphology, and given any syntactic structure *S*, there is some language *L* such that *O* has *S* in *L*. By itself, morphology imposes no constraints at all on syntax; it no more imposes constraints on syntax than it does on semantics. This is to be expected since syntax is, as we have seen, an aspect of semantics. The only way to have any idea what syntactic form an expression has, one must have a reasonably high-resolution knowledge of its semantics. To know the syntactic form of (11), one must know that 'white' is a predicate, 'snow' is a referring term, and so on. And to know that

12. 'Something is white'

can be inferred from (11), one must know (*inter alia*) the semantics of 'something'. Syntax-drivenness is a kind of semantics-drivenness: so syntax-drivenness is very much *not* innocent. Given a being that lacks a backlog of semantic knowledge, we can no more ask it to engage in syntax-driven operations than we can ask it to write poetry. So it makes dubious sense to regard syntax-driven operations as fundamental.

6. Tokens Versus Types

Sentence-types are abstract objects, and therefore do not have morphology. Sentence-*tokens* are what have morphology. The sentence-type 'I am tired' has no morphology. But any given token of that sentence has morphology.

But even though it is sentence-tokens, not sentence-types, that have morphology, two tokens of a given sentence-type can have absolutely nothing in common at the level of morphology. 'I am tired' can be tokened by noises, ink marks, hand signals, light beams, or smoke signals. No series of hand movements has any significant similarity to a deposit of ink.[12] Indeed, that sentence-type can be tokened by two deposits of ink that have very little in common. Consider the various different deposits of ink that token that sentence. Imagine all the different fonts and systems of short-hand that might be involved.

At the same time, two physical objects may be morphologically identical but token entirely different sentences (and thus have entirely different syntactic forms). There is some (possible) language in which an ink mark morphologically identical with 'Bob snores' means *if penguins liked classical music, they would prefer Bach to Beethoven* and thus has an entirely different syntactic structure from the one it has in English. Given any physical object *O*, no matter what its shape, and given any syntactic structure *S*, there is some language *L* such that, in *L*, *O* has *S*. This means that, by itself, morphology imposes no constraints on syntactic structure.

There is *a* connection between the syntax of an expression and its morphology. But that connection is indirect, and it lies in the *semantics* of the expression involved. Suppose that, when conversing with another English speaker, I make the noise 'Bob snores'. In that context, that noise has a certain syntax. But it has that syntax only because it has a certain semantics—only because 'Bob' is a singular term, '-s' is a tense-marker, 'snore' is a one-place predicate, and so on. If 'Bob' had not been a singular term, or 'snore' had not been a one-place predicate, then that noise would have had an entirely different syntactic form or no syntactic form at all. So *given* a semantics, there is, at least arguably, a connection between syntax and morphology. But absent a semantics, there is no such connection: in that case, syntax and morphology are completely independent of each other. To be responsive to the syntax of an expression, one must be alive to the relevant semantic facts. For reasons we've already discussed, this makes it implausible to suppose that syntax-driven operations have a foundational cognitive role.

7. Indexicality and the Dysmorphism between Syntactic and Logical Form

We have seen that the relation between syntax and morphology is, at best, extremely indirect. In this section we will see that, even if that relation were direct—even if it were one of identity—that would *still* not allow syntax-driven operations to have a foundational cognitive role. In connection with this, we will encounter new reasons to hold that logic-driven operations presuppose semantic, and extra-semantic, knowledge.

Let us start with some paradigms:

(i) Bob looks tired.
(ii) Bob looks ill.
(iii) Therefore somebody looks both tired and ill.

Consider (i). That sentence contains a tense-marker (the 's' on 'looks'). Tense-markers are indexicals. On Tuesday, you say 'Bob looks tired'. You have spoken truly; for at that time, Bob does look tired. On Wednesday, you say 'Bob does not look tired'. Again, you have spoken truly; for on Wednesday he does not look tired. Suppose that you didn't say anything between those two utterances. So you have said (albeit over a long period of time): 'Bob looks tired . . . Bob does not look tired'. You have not contradicted yourself at all. One could know everything there is to know about the *syntax* of your utterances during that period of time, and not have the slightest idea as to whether you had contradicted yourself. Granted, you said 'Bob looks tired. Bob does not look tired'. But you did not contradict yourself; and if someone were to think otherwise, she would be guilty of a grievous logical fallacy (Kaplan, 1979, 1989, pp. 586–587; Barwise & Perry, 1999, pp. 32–44).

Of course, there *are* cases in which an utterance of 'Bob looks tired . . . Bob does not look tired' is self-contradictory. But to know whether it is a self-contradiction, one has to know a great deal about the context in which both sentence-tokens occur. One has to know that they occur within a relevantly short period of time; one must know that the specious present picked out by the tense-marker in the first coincides, at least to a relevantly high degree, with that picked out by the tense-marker in the second. One must also know that the 'Bob' referred to in the second is the same as the 'Bob' referred to in the first. None of these facts can be read off of morphology; and, more surprisingly, none of these facts can be read off of syntax. One doesn't have enough *linguistic* knowledge to make a determination as to the validity or invalidity of that statement until one knows a fair amount about the contexts in which the expressions in question were tokened. None of this is narrowly syntactic knowledge (Barwise & Perry, 1999, pp. 43–44, for exactly this point).

What we've just said about (i)–(iii) is true of all sentences of natural language. Sentences of natural language typically contain tense-markers; so given only a knowledge of the syntactic structures of two sentence-tokens, nothing can be inferred as to the bearing that they have on each other. Even if, wholly on the basis of its morphology, one could know the syntactic structure of a sentence-token (or series thereof), that would *still* not typically suffice for a determination of its validity.

What we've said about tense-markers applies to indexicals that pick out places and individuals. If you are moving about and you say 'here is it warm; here it is cold; therefore there is some place where it is both warm and cold', you have made an erroneous inference, *unless* you know that both tokens of 'here' pick out the same place. And this knowledge is quite plainly semantic and, indeed, *extra*-semantic: it is, or involves, knowledge of the environment. It is easy to construct analogues of this point that relate to 'that', 'he', 'you' and most other indexicals.

The context-sensitivity of sentences prevents a mapping of syntax onto semantics. Logical form can be read off of syntax only in cases where the sentences involved are context-insensitive.[13] If our thinking consists in purely syntax-driven inferences, then those inferences will track logical form only if the expressions on which those operations are performed lack the kind of context-sensitivity just discussed. The expressions mediating our thought would therefore have to be stripped of context-sensitive components: they would have to be "eternal" sentences. But it is a point often made that a language lacking context-sensitive components is extremely *inefficient*.[14] Such a language is useless specifically when it comes to information relating to one's own welfare and to knowledge of one's environment: for such information is replete with references to what is here and now; and—valiant efforts notwithstanding—efforts to "eternalize" such information, to reduce indexical information to nonindexical information, simply don't go through (Barwise & Perry, 1999, pp. 32–34; Kaplan, 1989, pp. 557–558). There are many reasons for this. Here is one of them.

Let *L* be a language like English except that, in *L*, there are no indexicals that refer to the present time. In *L*, you have to have one expression for Monday, July 12, 2005, 2:05 p.m., and another word for Monday, July 12, 2005, 2:06 p.m. and so on. It is deeply unclear whether *L* would have the expressive power of English. Even if you are thirsty on Monday, July 12, 2005, 2:05 p.m., there is still a difference between thinking *I am thirsty now* and *I am thirsty on Monday, July 12, 2005, 2:05 p.m.* (Kaplan, 1989, pp. 557–558).

Now let *L** be a language that is like English except that, in *L**, *every* context-sensitive expression of English has been replaced with a context-insensitive one: *L** contains only expressions with fixed referents; it thus comprises only eternal sentences. For extensions of the reasons just given, *L** would have dramatically less expressive power than English. In large measure, languages are useful precisely because they contain context-sensitive expressions.

Also, it isn't even clear if languages like *L* and *L** can even exist; the existence of context-insensitive expressions seems to presuppose the existence of context-sensitive ones. English contains context-*in*sensitive terms that refer to specific times and places and individuals. We have expressions like '1492 A.D.', 'John Lennon', 'the place that is exactly 23 miles west of the western-most part of London', and the like. But suppose English didn't have any context-sensitive terms, and thus didn't comprise sentences like '*now* it is 2005', and '*that* guy is John Lennon'. Under that circumstance, it is hard to see how referents could be assigned to expressions like '2005 A.D', 'John Lennon', and other context-insensitive terms (Kaplan, 1989, pp. 557–558). Context-sensitivity is a pervasive, and fundamental, feature of language; and context-sensitivity weakens (and arguably severs) the connection between syntax and logical form (Barwise & Perry, 1999, pp. 32–34); and it makes logical form be even more of a semantic matter, as opposed to a narrowly syntactic one, than even we previously said.

In logic books, one encounters paradigms like:

(a) Socrates is a man.
(b) All men are mortal.
(c) Therefore, Socrates is mortal.

In this argument, the tense-marker appears to have little or no role; and, in this case, there is (at least arguably, though not in my view[15]) a straightforward mapping of syntax onto logical truth. The problem is that most sentences are not like this. If you pick an utterance of yours at random ('everybody's here', 'I don't think there will be a party tomorrow', 'you told me that Adam Smith never told his wife he loved her'), it will probably be replete with context-sensitive components; and, for reasons earlier discussed, semantics is needed to bridge the yawning gap between syntactic and logical form. The arguments in logic books are tendentiously chosen to lack such components.

If only for the sake of argument, let us suppose that our thinking does consist in syntax-driven operations on expressions; and let ''Mentalese'' (Fodor, 1975) be our name for the language to which those expressions belong. Let us evaluate this supposition in light of the points just made about context-sensitivity. If our thinking consists in purely syntax-driven operations on expressions, then *either* those operations will not be logic-tracking, for the reasons just discussed, *or* the expressions on which they are performed must lack the context-sensitivity characteristic of most natural language sentences. So CTM, in its need to have syntax track logic, requires that Mentalese sentences be frozen and context-insensitive—that they lack the context-sensitivity characteristic of their natural language analogues. But if Mentalese consisted entirely of frozen, eternal sentences, then it is hard to see how it could be of much utility to its user: what the organism needs to know is *specifically* indexical; it doesn't need (merely) ''eternal'' information; it needs egocentric knowledge of the here and now (Barwise & Perry, 1999, pp. 27–45).

To sum up, if Mentalese comprised only context-insensitive sentences, it would be incapable of mediating just the kind of information that the survival of the organism depends on. But the kind of context-sensitivity that enables a language to mediate such information is exactly the kind that blocks a purely syntactic characterization of logical truth in that language (Barwise & Perry, 1999, p. 45).[16]

8. Syntax as a Mirror of Semantics

These points about context-sensitivity give us crucial insight into an important matter. Our minds are content-driven (or, if you prefer, semantics-driven). In any case, our thought-processes are consistent with the representational contents of the thoughts they comprise. Suppose I think *Bob is in Greenland and Greenland is a cold place*. In virtue of that fact, I will probably form the belief *Bob is in a cold place*, but not *Jerry is in a cold place*. Any model of cognition must do justice to this fact. CTM (at least Fodor's version of it) accommodates this by taking the following line of thought:

> Syntax "mirrors" semantics. Our minds are syntax-driven. So even though our brains are not *directly* sensitive to semantics, they *are* directly sensitive to syntax. (For syntax ultimately reduces to morphology, and there is no difficulty explaining how a brain could be physics-driven.) And, although syntax and semantics are different things, nonetheless a kind of isomorphism can exist between them: semantics can be "coded" in syntax in roughly the way that spatial facts can be coded in arithmetical facts (in triples or four-tuples of numbers). Thus, the operations of a purely syntax-driven entity, like a brain, can be consistent with the *semantics* of the representations on which those operations are performed.[17]

As we've seen, given a language that contains context-sensitive features, there is no syntactic characterization of semantics, especially not of the semantic relations that matter most to CTM (those relating to proof, entailment and, more generally, relations of bearing). Let *L* be any such language. Let *S* be any sentence (or sequence of sentences) belonging to *L* such that *S* is logically true. There will be some sentence (or sequence thereof) S^* such that S^* has exactly the same syntax as *S* but such that S^* is not logically true. Consider the argument expressed by 'Socrates is a man; all men are mortal; therefore Socrates is mortal'. Because English includes tense-markers, and other indexicals, it is very easy to construct sentences that are syntactically indistinguishable from the one just given that express *fallacious* arguments:

> (*) 'He [pointing to John at time *t*] is tired; he [pointing to Fred at some other time t^*] is not tired; therefore somebody is both tired and not tired.'

(*) is doubly fallacious: it is nullified by the fact that the times in question don't coincide, and also by the fact that the demonstratives don't corefer. If thinking consists in syntax-driven operations, and therefore on expressions of some kind[18], those expressions must be replete with context-sensitive components, for the reasons earlier discussed. And this means that, in general, there is absolutely no syntactic characterization of the features of sentences involved in the determination of logical form; no syntactic characterization of entailment, probabilification, proof, and so on. So if we were syntax-tracking machines, we would be illogical except (at most) in the rare and special cases where context was irrelevant. It is probably appropriate to end this section by quoting Barwise and Perry (1999):

> There is . . . something profoundly misleading about the traditional concern over entailments between sentences. For one thing, their concern obscures the efficiency of language, and it is this efficiency that is central to meaning. For example the *sentence* SOCRATES IS SPEAKING does not follow from the *sentences* EVERY PHILOSOPHER IS SPEAKING [and] SOCRATES IS A PHILOSOPHER even though this argument has the same "logical form" (on most accounts of logical form) as [ALL MEN ARE MORTAL, SOCRATES IS A MAN, THEREFORE SOCRATES IS MORTAL]. In the first place, there is the matter of tense. At the very least the three sentences would have to be said at more or less the

> same time for the argument to be valid . . . . A rather startling consequence of this is that there can be no syntactic counterpart, of the kind traditionally sought in proof theory and theories of logical form, to the semantic theory of consequence. For consequence is simply not a relation between purely syntactic elements. (p. 22)

9. The Nature of Linguistic Meaning

For the sake of argument, suppose that thinking does consist in operations on expressions of some internal language, which we will continue to call "Mentalese." The expressions in question will not be sentence-*types*. Sentence-types are abstract objects. No brain state can constitute a sentence-*type*; for brain states are not abstract objects. A brain state can constitute, at most, a sentence-token. So if thought consists in operations on expressions, it consists in operations on sentence-*tokens*, not sentence-types. If I am not mistaken, this fact leads to a serious problem for CTM.

Under what circumstances does a physical entity qualify as a token of this, as opposed to that, sentence-type? Consider the *sound* 'snow is white'. If that sound were made by a volcano 50,000 years ago, no sentence-token would have been produced; nothing linguistic would have been accomplished. Consider the ink mark 'snow is white'. If, by some freak accident, a sand dune, on Neptune a million years go, were formed with that exact shape, no sentence would have been tokened; nothing linguistic would have occurred. So it is never merely in virtue of its physical form that a physical object qualifies as a token of this, that, or any sentence-type. Of course, form plays a part. *Part* of the reason my utterance of 'snow is white' tokens this, as opposed to that, sentence is that it has certain acoustical properties. But physical form is not the whole story.

Suppose that Bob, a cognitively normal speaker of English, produces the sound 'snow is white'. Here we have a genuine sentence-token. How is this case different from the ones just discussed? Bob made those sounds in a certain sociocultural context, and in consequence of certain linguistic intentions. Bob's noises qualify as a sentence-token not merely in virtue of their morphology, but also in virtue of their being embedded in a certain psychological, historical, and environmental context.[19] Those very sounds could mean anything or nothing. The reason they mean *snow is white* is entirely that they occur in a specific context.

A corollary is that to have any idea what that sound means, you must know something about the context in which it occurs. Absent such knowledge, those sounds are, from your viewpoint, comparable to the sand dune and the volcano-noise we discussed earlier. Under that circumstance, those sounds will have no *logical* form at all, at least not from your viewpoint. Responsiveness to logical form presupposes linguistic knowledge; and linguistic knowledge presupposes the contextual knowledge just described.[20,21]

A genuinely linguistic response to a physical occurrence involves knowledge of a doubly conditional proposition: *if* an occurrence with such and such morphological properties takes place, *then* that occurrence has such and such meaning, *provided* that it occurs in a certain kind of context.

For the sake of argument, suppose that thinking consists in operations on sentence-tokens. For reasons just discussed, if your brain is to respond to the logical forms of those sentence-tokens, then it must your brain must know a fair amount about the facts surrounding the occurrence of those tokens. Such operations presuppose extra-linguistic knowledge, and therewith a wealth of cognitive and representational abilities. So it makes little sense to suppose that such operations could be foundational.[22]

10. The Concept of an Algorithm

The idea that syntax-driven operations might be cognitively foundational has been encouraged by a kind of naiveté regarding concepts like ALGORITHM, FORMAL INFERENCE, MECHANICAL PROCEDURE, and by a subsequent tendency to mistakenly lump these terms together.

There are algorithms for long division, multiplication, addition, and so on. Let us consider the algorithm for addition (i.e., for the addition of multidigit numbers). Here is a widely held conception as to the nature of that algorithm: it is a conception I wish to combat.

> Consider the expressions '172' and '35'. There is an algorithm that enables you to add these terms together, and thus to generate a theorem of number theory, *without* knowing anything about the meanings of the expressions involved, and without having prior knowledge of number-theoretic truth. You line up the '172' and '35' in a certain way. You thus form columns of number-expressions. The algorithm tells you to write an expression with a certain shape underneath those columns: what you are supposed to write is a function entirely of the shapes of the expressions in each of those columns. At no point is any judgment or acumen required: you need only be sensitive to the shapes of the expressions involved. If you follow this algorithm—if you write the right shapes in the right places—you will produce a correct number-theoretic statement. A machine could be designed that wrote the right shapes in the right places, and that therefore executed this algorithm. Therefore a machine could be designed that did addition and, consequently, that generated correct number-theoretic statements.

There is a serious problem with this view. Any expression—any inscription, any noise or pattern of light—could mean anything. In English, '172' denotes the number 172. But it could just as well denote Socrates or the number five or nothing at all. Given any entity at all, there is some language *L* such that, in *L*, '172' denotes that entity. Since any shape can be associated with any meaning and can thus have any truth-value, it follows that there is no purely morphological characterization of truth, even for lowly domains like addition. There can be no algorithm that enables you (or any other entity) to decide the truth-value of an inscription *wholly* on the basis of its shape. Truth-value can be decided, if it all, only on the basis of morphology *and* the semantic background.[23]

Suppose that '1' meant *Plato*, and '2' meant *elephant*, but that otherwise our semantic conventions were unchanged. In that case, '172 + 35 = 207' would be false or nonsensical. There are infinitely many (readily constructed, if not currently existing) languages such that, in them, '1' means *Plato*, '2' means *elephant*, and so on. The algorithm for addition assigns truth to (e.g.) '172 + 35 = 207' only *if* that inscription is part of a very specific semantic system. So that algorithm is given by—is identical with—a conditional proposition: *if* '1' denotes 1, 'two' denotes two, '+' denotes addition, and so on, *then* there is a function assigning truth to certain patterns of those expressions, and falsity to others, on the basis of the morphologies of those patterns. To apply that algorithm is to recognize one of the consequences of that conditional proposition; more exactly, it is to recognize the bearing of that conditional proposition on the question whether a particular inscription[24] (e.g., '172 + 35 = 207') is true or (what is closely related) on the question how a particular inscription (e.g., '172 + 35') must be completed ('= 207') if a true expression is to result. So to apply that algorithm involves recognizing the truth of a *multiply* conditional proposition, one that is at least roughly like this: Given that '1' denotes 1, '2' denotes 2, '+' denotes addition, and so on (in other words, given that such and such shapes have such and such meanings); and given also that, relative to that semantic system, there is a function from such and such shapes to such and such truth-values; and given finally that '172 + 35 = 207' has such and such shape; *then* it follows that '172 + 35 = 207' has a certain truth-value.[25]

So using the algorithm for addition involves, among many other things, one's taking that inscription to be part of a certain semantic system; it thus involves background semantic (and extra-semantic) knowledge, and is thus not purely "mechanical."

As we noted, given any pattern of shapes, there is some language *L* such that, in *L*, that pattern is meaningless or false. In light of this, consider an arbitrary act of shape-manipulation. (But focus on the shape-changing itself: leave out any semantic or extra-semantic knowledge that might lie behind it.) For every language *L* in which that act of shape manipulation constitutes (or generates) a true statement, there are infinitely many in which it constitutes nonsense or falsity. So to compute is not (just) to manipulate shapes. The manipulation of shapes is, at most, a causal consequence of an actual computation—of one's recognizing the truth of a hyper-conditional proposition of the kind earlier described. The manipulation of shapes could almost be described as a kind of epiphenomenon.[26]

These considerations show that, even if there algorithms for number-theory, history, economics, and so on, that would still not mean that number-theoretic, historical, or economic *thought* could be "mechanized." Applying an algorithm is not a purely mechanical procedure; it is a judgment of a certain kind; and it involves, among other things, background semantic knowledge.[27]

"But surely," we are told, "an algorithm is a mechanical procedure, and the application of an algorithm is a mechanical process." The statement 'the execution of an algorithm is a mechanical process' is ambiguous. There is a *certain* truth in each disambiguation. But, strictly speaking, neither disambiguation is quite accurate.

Remember that an algorithm is given by a conditional proposition: *if* such and such a semantic system holds, *then* (relative to that semantic system) there is a morphological characterization of truth (i.e., a function from shapes to truth-values). To apply an algorithm, as we saw, is to recognize the bearing of this conditional proposition on the question whether some inscription (or noise . . .) is true or false. So there are two components to the application of an algorithm. The first component consists in a recognition of the operativeness of some semantic system. The second component involves the existence of patterns of ink (or light or sound . . .) having the right shapes. The *second* component can be realized mechanically: it can be realized by a being that hasn't even a flicker of awareness of any kind—by windmills, punch-card automata, what have you. But the first component is not in any sense mechanical.

When we say that "computing," and "applying an algorithm," are "mechanical" procedures, that is either false or it is an abbreviated way of saying that the *second* of the two components of applying an algorithm can be realized mechanically (without any sensitivity to semantics, without any judgment).

There is another, slightly different (but related) interpretation of the statement 'the execution of an algorithm is a purely mechanical affair'. Suppose you want to figure out some sum, say the sum of 172 and 35. There are (at least) two different ways you could go about doing this. You could reflect very hard on the concepts involved (172, +, and so on); and, on the basis of a scrutiny of those concepts, you could ascertain that 172 added to 35 must equal 207. Or you could use a procedure that is much more energy-efficient (and intelligence-efficient). Let us discuss this second procedure. First of all, we obviously have semantic conventions regarding these numbers. These conventions are to the effect that '1' denotes one, '+' denotes the operation of addition, and so forth. Let us refer to the set of these conventions as *S*. So *S* is a kind of language. You know that if you write down the *S*-expressions denoting 172, 35, and so forth, and you make sure that they are lined up in the right way, then there is a rule telling you how to complete that shape—a rule telling you how to produce a shape that, relative to *S*, constitutes a true statement regarding what happens when 172 is added to 35. If you take advantage of this rule, you will be spared a great deal of thought that you wouldn't otherwise be spared.

This doesn't mean that *no* cognition is involved. Obviously *some* kind of cognitive processing is involved. You use the information given you by sight (or touch or, conceivably, other sensory modalities) to make sure that the ink marks have the right shape and are lined up the right way. You also call on your memory; you remember what kind of shape you're supposed to write in what place. Also, you know how to interpret, and symbolize, the shapes. You know that '1' denotes 1, and so on. This knowledge is involved in your symbolizing the operation in question (that of adding 172 to 35) in such a way as to render applicable the rule you wish to exploit. (If you don't know how, in *S*, the number one is supposed to be written down, then the aforementioned rule isn't going to be of much use to you; you won't be able to apply that rule, to exploit it to perform a computation.) So even when you add in the "quick and dirty way"—i.e., even when you line up the '172', with the '35' (in the right way), and so on, and then dutifully crank out an '= 207'—cognition is still involved: sense-perceptions are used, memories of semantic conventions are involved. So that act is by no means mechanical. But there is, granted, *a* sense in which that act is much *more* mechanical—much *less* thought-heavy—than the alternative procedure (i.e., the procedure of reflecting deeply on the number-concepts involved).

(To facilitate further discussion, let us systematically use the term 'the quick and dirty way' to refer to the kind of operation that we've just been discussing: the kind where one lines up inscriptions with such and such shapes, in such and such a manner . . . the kind where *mathematical* reflection is minimized.)

That alternative procedure—the *non*-quick and dirty method—involves a fair amount of *mathematical* thought. The quick and dirty procedure involves very little mathematical reasoning. The only cognition involved in the quick and dirty procedure is "easy": it involves processing that anyone (who is not severely damaged) can do "automatically": the production of ink marks having certain shapes, the lining up of those ink marks in a certain way, the remembering of some fairly simple symbolic conventions, the sense-perceiving and manual production of certain marks. It is wrong, as we've seen, to describe as "purely mechanical" (if by that is meant cognition-free) the quick and dirty method of adding. But it is *not* entirely wrong to describe that process as *automatic*. In some sense of the word 'automatically', you can automatically drive a car, turn on a light, understand a sentence of your native tongue. But that doesn't mean that no cognition is involved, that no processing of representations is involved. When you stop at a traffic sign, or understand a sentence of your native tongue, the operations involved are replete with content; they are by no means representationally or cognitively innocent. Perceptions are involved; information is uploaded from those perceptions (that information-uploading is itself, if cognitive scientists are right, a cognition-heavy matter); that information interacts with background knowledge (e.g., *you must stop at red lights or you will be given a ticket or jeopardize your safety . . .*). These things—stopping at traffic signals, understanding a salutation in your native tongue—are "automatic," on some interpretation of the word. But they are not cognition-free.

We tend to run together terms like 'automatic' and 'mechanical'. So perhaps in *some*, extremely loose, sense of the word 'mechanical', those sorts of operations are "mechanical." But they are by no means cognition-free. Far from it. When you add numbers, using the quick and dirty method, your performance is automatic and thus, in a very loose sense, "mechanical." But it is by no means cognition-free; and it presupposes an enormous background of knowledge, representational abilities, and information-processing power.[28]

Let us sum up. Sometimes when it is said that adding is (or can be) a "mechanical" process, what seems to be meant is that it is cognition-free and representationally innocent. But *that* statement, we have seen, is absolutely false. As we just noted, when a person adds numbers—even if she uses the quick and dirty method—various background representations and capacities are mobilized. On the other hand, when a representationally innocent entity generates a pattern of shapes, that *by itself* does not constitute an act of addition; it constitutes only one of two components of such an act. The other component involves the recognition (on somebody's part) that *relative* to some operative semantic system, the aforementioned pattern of shapes constitutes a true statement. Under no circumstances is a genuine case of adding a representationally, semantically, or cognitively innocent matter; under no circumstances is it "mechanical" in any strict sense.[29]

We should deal with a possible objection to our general line of thought:

> Surely there is a sense in which a Turing machine computes. And surely such a thing computes by scanning physical objects—by responding in the right way to their morphologies. Surely what we just said about Turing machines is true (mutatis mutandis) of computers generally. Given that there are such things as computers, it is clear that there is something wrong with your arguments.

If 'compute' means 'simulate logic-driven operations', then it is indisputable that various artifacts compute. But what we want to know is: Do such artifacts compute in the sense that they *actually* engage in logic-driven operations? *That* is the sense of 'compute' that is relevant to discussion of the foundations of mind.

As we've seen, to engage in a logic-driven operation, a thing must be sensitive to semantics: it must *already* have a great deal of semantic and, indeed, extra-semantic knowledge. I've never heard anyone, whether an advocate of CTM or not, say that Turing machines *do* respond to semantics. Given that the Turing machine isn't responding to semantics, it follows that it isn't computing (in the relevant sense): computing involves being sensitive to logical form; and logical form is buried in semantics (compare Wittgenstein, 1983, p. 257).

As we have seen:

- There is no purely morphological characterization of syntax.
- There is no purely syntactic characterization of logical truth.
- The only adequate characterizations of logical truth are therefore, at least partially, semantic.
- The only way to sensitize a thing to logical form is to sensitize it to semantics.

Since the Turing machine is not semantics-driven, it is not responding to logical form. Therefore it is not computing (in the relevant sense),[30] even though, from some viewpoint, it acts like something that is computing (Coulter, 1983, pp. 65–68; Wittgenstein, 1983, pp. 257–258, 422).[31]

Notes

[1] This is arguably subject to some qualifications associated with externalism. But these will not come into play in the present paper.

[2] Once again, this is arguably subject to some qualifications relating to epistemic externalism that won't play a significant part in the present discussion.

[3] One *conceivable* way out is to say that our mental processes are over-determined: they are determined *entirely* by the laws of physics (for they are realized by brain events, and brain events are determined by the laws of physics). But they are *also* determined by consciousness

of norms. But this is a radically implausible thesis. Let *E* be an arbitrary brain event that is identical with, or that realizes, a mental event. For the sake of argument, suppose that *E* is "over-determined" in the relevant sense. This would mean that *E* was *entirely* caused by some state of affairs *S* (some purely physical or physiological state of affairs) and *also* that *E* was caused by some state of affairs *S** (some awareness of the canons of rationality), such that *S* and *S** were completely distinct. But the idea of an event's being *entirely* caused by two distinct states of affairs is of doubtful coherence. Also, if *each* brain event that realized a mental event were *entirely* caused *both* by some awareness of norms and *also* by some physiological state of affairs, then we'd have innumerable cases of two entirely different kinds of things eventuating in the exact *same* thing: we'd have innumerable cases where two entirely distinct things—one a consciousness of norms, the other a physiological state of affairs—independently eventuated in the exact *same* thing (some brain event). But to accept this is to accept innumerable unexplained coincidences. By contrast, if we suppose that mental states of affairs *are* neuro-cerebral states of affairs, then it is no longer a coincidence that brain events realizing mental entities have physiological causes *and* mental causes: the former *are* the latter, at least in some causes. So it would not be good methodology to accept the idea that brain states are over-determined. And practically no one does accept it. (I myself do not.)

[4] I should make one point very clear. I am not opposed to the idea that *some* thought consists in computation. I could perform a computation right now. And I am not opposed to the idea that many of our cognitive achievements derive from subpersonal acts of computation (like those posited by Chomsky and Marr). What I am opposed to is the idea that computation might constitute the *foundation* of cognitive activity. Computation, I will argue, exists only in a rich background of cognitive and representational accomplishment. So even though computation might be a pervasive feature of our cognitive lives, it cannot be the *basis* of it; it necessarily has a subordinate (though perhaps deeply important) role in cognition.

[5] Fodor (1981): "Computational processes are both *symbolic* and *formal*. They are symbolic because they are defined over representations, and they are formal because they apply to representations in virtue of (roughly) the *syntax* of the representations" (p. 226). A few paragraphs later, Fodor writes: "Formal operations apply in terms of the, as it were, shapes of the objects in their domain" (p. 227). This passage contains the hedge "as it were." But on any interpretation of this hedge, as we will see, there simply doesn't exist the kind of connection between syntax and morphology (shape) that Fodor's position alleges.

[6] See Fodor (1987): "The operations of the machine [on which I am modeling the human mind] consist entirely of transformations on symbols; in the course of performing these operations, the machine is sensitive solely to syntactic properties of the symbols; and the operations that the machine performs on the symbols are entirely confined to altering their shapes" (p. 19). See also Fodor (1990): "You connect the causal properties of a symbol with its semantic properties via its syntax . . . . To a first approximation, we can think of its syntactic structure as an abstract feature of its (geometrical or acoustic) *shape* [original italics]. Because, to all intents and purposes, *syntax reduces to shape* [italics added], and because the shape of a symbol is a potential determinant of its causal role, it is fairly easy to see how there could be environments in which the causal role of a symbol correlates with its syntax. It's easy, that is to say, to imagine symbol tokens interacting causally *in virtue of* [original italics] their syntactic structures. The syntax of a symbol might determine the causes and effects of its tokenings in much the way that the geometry of a key determines which locks it will open. But, now, we know from modern logic that certain of the semantic relations among symbols can be, as it were, "mimicked" by their syntactic relations; that, when seen from a very great distance, is what proof-theory is about. So, within certain famous limits, the semantic relation that holds between two symbols when the proposition expressed by the one is entailed by the proposition expressed by the other can be mimicked by syntactic relations in virtue of which one of the

symbols is derivable from the other. We can therefore build machines which have, again within famous limits, the following property: The operations of the machine consist entirely in transformations of symbols; in the course of performing these operations, the machine is sensitive solely to syntactic properties of the symbols; and the operations that the machine performs on the symbols are entirely confined to altering their shapes. Yet the machine is so devised that it will transform one symbol into another if and only if the propositions expressed by the symbols that are so transformed stand in certain semantic relations—e.g. the relation that the premises bear to the conclusion in a valid argument. Such machines—computers, of course—just *are* [original italics] environments in which the syntax of a symbol determines its causal role in a way that respects its content. This is, I think, a pretty terrific idea; not least because it works". (p. 22). Contrary to what Fodor holds, we will see that building a syntax-driven machine is as difficult as building a semantics-driven machine. For syntax is an aspect of semantics; and the connection between the morphology of a symbol and its syntax is as remote and conditional as the connection between a symbol's morphology and its semantics. See also Fodor (1981, pp. 13–17, 1987, pp. 18–20, 1990, 1994, pp. 294–298) and Pylyshyn (1984).

[7] Expression-tokens can be spoken or realized by hand signals or light beams. They needn't be realized by ink marks or by anything else that has "shape" in the narrow sense. In this paper, I will use the term 'morphology' in a generalized sense—to describe acoustical (or hand signal) *counter-parts* of the physical shape had by written expression-tokens.

[8] Compare Fodor (1981, pp. 226–227). Having written that "[mental processes] are formal because they apply to representations in virtue of (roughly) the *syntax* of the representations"—in other words, having identified "formal" with "syntactic" operations—Fodor then goes onto write: "Formal operations are the ones that are specified without reference to such semantic properties of representations as, for example, truth, reference, and meaning." This, I will argue, is dead wrong. Formal, in the sense of syntactic, operations are a kind of semantic operation.

[9] I should reiterate that this is in no way meant to conflict with the important discoveries of Chomsky and Marr. Obviously people *can* compute; and people may well engage in the subconscious (and subpersonal) computations posited by Chomsky and Marr. So in *that* sense, computations may have a very important role in our cognitive lives. What I am saying is that, if Chomsky and Marr are right (as they may well be), the computations they posit presuppose an even more fundamental stratum of cognitive functioning: for any computation, in a relevantly robust sense of the word, presupposes a considerable cognitive and representational background, as I will argue.

[10] Quine (1970) wrote: "[A] logical truth is a truth that cannot be turned false by substituting lexicon [lexical items] for lexicon" (p. 58). In other words, a logical truth is one that cannot be made false by replacing *lexical items* with *lexical items*. And a "lexical item" is defined as one that *picks something out* or, at any rate, is *supposed* to pick something out, given its grammatical role. ('Socrates' and 'fat' are lexical, because they pick out an individual and a property, respectively. 'Pegasus' is lexical because it is *supposed* to pick something out; its grammatical role is like that of 'Socrates'.) Carnap (1947, pp. 85–88) described any sign that was "descriptive"—i.e., that picks something out or expresses a property—as being "nonlogical." In his view, purely "logical" terms—by which he means 'or', 'not', 'and' and so on—do not pick out or describe or express anything. They are purely "formal." We will take issue with this view in due course.

[11] These ideas are found, in some form, in Russell (1905/1990). (Russell himself denies that quantifiers refer. But much of the contemporary inspiration for the view that they refer to second-level functions is due to Russell.) Ultimately, these ideas go back to Frege (1892a, 1892b). Richard Montague (1974) used them to great advantage in his work on natural language quantification. See McCawley (1981, ch. 14) for a brief, clear discussion of some of the relevant aspects of Montague's work.

[12] There is *a* sense in which a spoken of token of 'I am tired' is isomorphic with a written token and with a token in hand signals. But *that* sense of 'isomorphic' does not have to do with *morphology*; it has to do with semantics. *Given* that any two of those three tokens have certain *semantic* similarities, the written 'I' can be put into a certain kind of correspondence with its spoken (or hand-signaled) counterpart, the same being true (*mutatis mutandis*) for the spoken 'tired', the spoken 'am', and so on. But *absent* that semantic similarity—if we just focus on physical morphology—those three tokens have about as little in common as any three things. Any ''isomorphism'' that holds between them is underwritten by semantic similarities, not (mere) morphological similarities.

[13] This point is clearly made by Barwise and Perry (1999, pp. 32–44). It is also made by Kaplan (1989). It ultimately goes back to Strawson (1951).

[14] Barwise and Perry (1999, pp. 32–39) deserve credit for making this point very clearly; they used the term 'efficiency' to mark the feature of language which I am discussing.

[15] My own view is that, even here, there is no purely syntactic characterization of logical truth. It is true that, in (a)–(c), the indexicals are not important. But the *reason* they are not important has to do with the peculiarities of the concepts involved—it has to do with the fact that, whether members of a species are mortal is not a function of time: if members of a species are mortal at *t*, they will be mortal at time *t**. That fact, in its turn, turns on peculiarities of the property of mortality. Not all properties are like that. For example, whether all members of a species are thirsty or are in Antarctica *is* a function of time. The relevant differential facts about the concept of mortality are not coded in the *syntax* of (a)–(c). So the irrelevance of the tense-markers in (a)–(c) is a function of subtleties buried in the *semantics* of the terms involved in that argument. So even where artifacts like (a)–(c) are concerned, there is a gap between syntax and logical form, and that gap is bridged only by semantics.

[16] Conceivably, one might take a line like the following:

> Consider the sentence-token: (*) 'Bob now looks tired, and Bob now does not look tired'. The syntactic form of (*) is a function of whether the two occurrences of 'Bob' corefer; and it is a function of whether the indexicals ('now', '-s') refer to the same bit of time. If the expressions in question *don't* corefer, (*) has one syntactic form; if they *do* corefer, it has some other syntactic form. So the facts you mention about indexicality, and about ambiguity, do *not* create a gap between syntactic and logical form. They show only that syntactic form is more fine-grained than you allow. First of all, this response is completely ad hoc, and is contradicted by everything we know about the concept of syntax.

But even if correct, this response isn't available to CTM; for it amounts to saying that syntax is even *more* a matter of semantics than even we previously said. The objector is saying that the very *syntax* of (*) is a function of the specific referents of the indexicals, and of the singular terms, in it. So, if the objector is right, to know the syntactic form of (*), it is not enough to know generic facts about the semantics of its expressions; it is not enough to know that 'Bob' is a singular term, and that each occurrence of 'now' refers to time of utterance: one has to know to *which* specific person 'Bob' refers, and what the exact times of utterance are. The objector's response thus makes syntax be totally inaccessible to anything that lacks fairly high-resolution knowledge of semantics; it thus makes syntax-drivenness *presuppose* the existence of various representational and cognitive abilities. Thus the objector's point makes it even harder for syntax driven operations to form the foundation of our cognitive lives.

[17] See Fodor (1994): ''We know from formal logic that certain of the semantic relations among symbols can be, as it were, 'mimicked' by their syntactic relations; that, when seen from a very great distance, is what proof-theory is all about. So, within certain famous limits, the semantic relation that holds between two symbols when the proposition expressed by the one is implied by the proposition expressed by the other can be mimicked by syntactic

relation in virtue of which one of the symbols is derivable from the other. We can therefore build machines which have, again within famous limits, the following property: the operations of such a machine consist entirely of transformations of symbols; in the course of performing these operations the machine is sensitive only to syntactic properties of the symbols; and the operations that the machine performs on the symbols are confined entirely to alterations of their shapes. Yet the machine is so devised that it will transform one symbol into another if and only if the symbols so transformed stand in certain *semantic* relations; e.g. the relation that the premises bear to the conclusion in a valid argument. Such machines—computers, of course, just *are* environments in which the causal role of a symbol is made to parallel the inferential role of the proposition it expresses'' (pp. 22–23). Fodor hedges his points with phrases like: ''within certain famous limits,'' ''as it were,'' ''from a very great distance,'' and ''again within famous limits.'' These hedges turn out either to be empty or to conceal systemic problems with Fodor's analysis. The ''famous limits'' Fodor has in mind have to do with incompleteness results, the most famous being the two of Gödel's. An apparent (and, in the view of most, an actual) implication of these results is: the bearing relations that hold among number-theoretic statements cannot be reconstituted within the confines of the concepts corresponding to so-called syntax-defining terms like 'not', 'or', 'and', 'for some', and the like. Admittedly, the implications of such incompleteness theorems are a matter of great controversy. But such exotic results in mathematical logic are the *least* of the problems affecting a mirroring of semantics in syntax. Remember what we said earlier. The syntax of a sentence is a function of the more generic facts about its semantics. 'Bob is tired' has a certain syntax because 'Bob' is a singular term, 'tired' is a predicate (a one-place function), and so on. For this reason 'Tim is angry' and 'Bob is tired' have the same syntax. Logicians have created (or outlined) languages in which, supposedly, logical form is made perspicuous by syntactical structure. But no matter how perspicuous a language is (from the view point of its syntax reflecting its semantics), sentences having very different semantics will have the same syntax. In any language having any expressive power at all, there will be semantically different sentences having the same syntax. So a mind that tracks syntax will inevitably fail to do justice to semantic relations. (Henceforth, to facilitate discussion, sentences in boldface are supposed to be the Mentalese translations of their English counterparts.) Fodor would not necessarily be moved by these points. For he seems to have held that '**Mary is nervous**' and '**Fred is angry**' *do* have different syntaxes. His reason, if I understand him correctly, is they have different *morphologies* and morphology is, or reduces to, syntax. (This point is crucial to Fodor's way of dealing with Frege-cases and also with the apparent truth of externalist theories of content, like Putnam's. See Cain, 2002, p. 135; Fodor, 1994.) But, as we've seen, morphology doesn't have *remotely* the kind of connection to syntax that it would have to have for '**Mary is nervous**' and '**Fred is angry**' to have different syntaxes. And, plainly, they have the same syntax, the reason being that they are like in the relevant semantic respects ('**Mary**' and '**Fred**' are both singular terms, '**angry**' and '**nervous**' are both one-place predicates, and so on). Perhaps Fodor can circumvent this problem. But it is quite clear that, for any one syntactic form, there are many sentences having that form that have different semantics. And this seriously problematizes the idea, so crucial to CTM, that semantics can be mirrored in syntax. In response, one could say (in effect, Fodor, 1994, *does* say) that the morphological differences between '**Mary is nervous**' and '**Fred is angry**' give them different causal roles, explaining why they lead to different beliefs. But this leads back to the problem we were discussing earlier (pages 5–7). A morphology-driven operation is not cognitive at all—is merely *simulated* thought— except in so far as syntax (or semantics) is realized by morphology. But morphology, by itself, to no degree realizes syntax (or semantics), as we've seen time and time again. So if the differences in causal role between '**Mary is nervous**' and '**Fred is angry**' are purely morphology-based, then those differences are not *cognitive* at all. Quite plainly, there is a *cognitive* basis to my willingness to infer *somebody is nervous or tired* from *Mary is nervous*, but not from *Fred is angry*. So if, as Fodor supposes, our thoughts are mediated by Mentalese expressions, then there is a *cognitive* basis to the different causal roles had by '**Mary is nervous**' and '**Fred is angry**'. So the difference between '**Mary is nervous**' and '**Fred is angry**' *cannot* be purely morphology-based. Conceivably, Fodor could counter-reply by saying this: For the reasons you give, where symbols of natural language are concerned, any shape can be associated with any syntax at all, given the right background facts. But things are different where Mentalese symbols are concerned. In Mentalese, morphology *does* realize syntax. Since, as we have seen, syntax is an aspect of semantics, this reply amounts to saying: "In Mentalese, morphology realizes semantics." But no advocate of CTM, and definitely not Fodor, would say that morphology realizes *semantics*, even where Mentalese is concerned. CTM is a form of functionalism. Functionalism says that the semantics of a brain state supervenes *not* on its morphology but on its causal role. So given that syntax supervenes on semantics, functionalism says, in effect, that the *syntax* of a brain state does not supervene on its morphology. (In many places, Fodor vehemently argues that the semantics of brain states don't supervene on *any* of their intrinsic properties, let alone on their shapes. So Fodor himself would never say that, where Mentalese is concerned, semantics supervenes on morphology. See Fodor, 1981, 1994, 1998.)

[18] Fodor (1994) writes: "There must be mental symbols because, in a nutshell, only symbols have syntax, and our best available theory of mental processes . . . needs the picture of the mind as a syntax-driven machine" (p. 23). Fodor is quite right to say that only linguistic expressions have syntax. So he is quite right to say that *if* thinking consists in syntax-driven operations, *then* there is an internal language (a "language of thought"). But, unfortunately for Fodor's position, syntax doesn't adequately reflect logical form, for the reasons we've just been considering. So if we were syntax-driven machines, we would be logic-blind.

[19] This point, or one like it, is famously found in Wittgenstein (1958), though Wittgenstein's point seems to be less mentalistic. Similar points are found in Putnam (1975) and Burge (1979).

[20] I would like to thank an anonymous review at Philosophical Psychology for inducing me, through her/his sharp comments, to make the following points. Since Frege, it has been generally assumed that, in natural language, syntax is a poor indicator of logical form. Consider: (a) 'Bill is a green duck, therefore Bill is a duck' (b) 'Bill is a decoy duck, therefore Bill is a duck'. The stock line on this sort of phenomenon is as follows.

> In (a), syntax tracks logic; in (b), syntax does not track logic. But (a) and (b) have the same syntax. Therefore syntax does not track logic, at least not in natural language. By itself, this is not necessarily a problem for CTM. CTM can say: Thought consists in syntax-driven operations; but those operations involve symbols of what we might call a "well-behaved" language—the kind that Frege and Russell talked about: a language in which there is a kind of coincidence between logical and syntactical form. In English 'Bill is a decoy duck' has the same syntax as 'Bill is a green duck'. But 'Bill is a green duck' means *Bill is a duck and Bill is green*, whereas 'Bill is a decoy duck' does not mean *Bill is a decoy and Bill is a duck*. If thinking consists in syntax-driven operations, the medium of computation is a language in which doesn't contain distortion like 'Bill is a decoy duck'—everything is made perspicuous.

This response, though superficially reasonable, leaves some important issues untouched. No speaker of English would regard (b) as a legitimate inference. Why not? Because anyone who knows the meaning of (b) knows that it is not equivalent to: 'Bill is a duck and Bill is a decoy, therefore Bill is a duck'. A knowledge of *meaning* is what prevents faux-inferences like (b). This point actually applies even to the idealized languages of which the objector speaks, and can be used to block the objector's point. Suppose we spoke a language—call it English*—that is like English except that, in English*, syntactical form mirrors logical form. So the English*-translation of 'Bill is a green duck' is 'Bill is green and Bill is a duck'. And the English*-translation of 'Bill is a decoy duck' is *not* 'Bill is a decoy and Bill is a duck'; it is (roughly) 'Bill is an object that looks like a duck, but is not a duck'. So English* doesn't contain the devices that, in English, make syntax diverge from logical form. We know that, in *actual* English, knowledge of semantics is needed to avoid faux-inferences like (b): if one relies only on syntax, one is thrown off. But the same is true of English* no less than English. Consider the English*-sentence: (c) 'Bill is green and Bill is a duck, therefore Bill is a duck.' You couldn't possibly know what syntax (c) had unless you knew that 'Bill' were a singular term (i.e., denoted an individual), 'green' were a one-place predicate (i.e., denoted a function of one variable), and so on. If these terms had different denotations, then (c) would have an entirely different syntactic form. So even in English*, syntax-driven operations—so-called "formal" inferences—are semantics-driven. The difference between English and English* is not that, in the latter one needs *no* semantic knowledge to make formal inferences, whereas, in the former, one does need such knowledge. The difference lies in how much semantic knowledge is needed. In English, one needs knowledge of the specific meanings of predicates to make the right inferences. In English*, one needs only fairly generic knowledge; one doesn't need to know the *specific* meanings of expressions; one need only know that (e.g.) 'small' denotes *some* one-place predicate or other. But in any language, no matter how well-behaved, syntax-driven operations are semantics driven. When you say that a language is "well-behaved" (in the sense earlier defined), you aren't saying that, in it, the set of semantic facts is reflected in *non*-semantic facts; you are saying that the set of semantic facts is reflected in a certain subset of that same set. In many places, Chomsky has argued compellingly that 'John expected Bill to leave' and 'John persuaded Bill to leave' have very different syntactic structures, even though, from the viewpoint of the grammar we learned in school, they have the same syntactic structure. If Chomsky is right, the syntactical differences between them turn on subtleties about the *semantic* differences between 'persuade' and 'expect'. So syntax turns out to be buried in semantics.

[21] To block this line of argumentation, CTM must say this: Where symbols of natural language are concerned, any shape can be associated with any syntax at all, given the right semantic background. But things are different where Mentalese symbols are concerned. In Mentalese, morphology *does* realize syntax.

[22] But, as we saw in note 17, this line of thought is probably not available to CTM. Advocates of CTM often talk about how semantics can be "coded" in morphology; and they say that, because Mentalese semantics is appropriately "coded" in Mentalese morphology, morphology-driven operations on such symbols constitute content-driven operations. The word 'coded' here is ambiguous; and neither disambiguation is favorable to CTM. On one interpretation, to say that Mentalese semantics is "coded' in Mentalese morphology is to say that there is a correspondence—a one-one function (or, at any rate, some kind of function)—from Mentalese semantics to Mentalese morphology. As we just saw, in *this* sense of the term 'coded', given only that Mentalese semantics is coded in Mentalese morphology, it does *not* follow that a morphology-driven operation on Mentalese symbols is a computation or inference of any kind. What *drives* the play of symbols must be a *knowledge* of that correspondence, a *knowledge* of that coding-relation; and such knowledge is obviously semantic (and, indeed, extra-semantic). On another interpretation of the word 'coded', to say that Mentalese semantics is "coded" in Mentalese morphology is to say that Mentalese morphology *is* Mentalese semantics or, in any case, that Mentalese semantics supervenes on Mentalese morphology. But, as we saw in notes 17 and 22, it is probably not open to CTM to say that Mentalese semantics supervenes on Mentalese morphology.

[23] Of course, where non-analytic statements are concerned, knowledge of the *extra*-semantic is also required.

[24] Or other physical entity, e.g., sound or pattern of light.

[25] I don't know whether, at the given level of detail, this characterization is accurate. But my point is that applying that algorithm consists in making a judgment, in recognizing the truth of a proposition, indeed an extremely conditional one. It may turn out—probably will turn out—that the proposition in question is even *more* complicated and conditional than we

have here said. But that would only confirm my point that to apply an algorithm, to compute, is to recognize the truth of a proposition, and is not to manipulate shapes.

[26] Even the use of truth tables to perform a computation is not, strictly speaking, a mechanical process. For every language in which a given shape—say, '*P* or not *P*'—expresses the disjunction of a proposition and its negation, there are infinitely many in which it expresses the *conjunction* of a proposition and its negation, and infinitely many in which it expresses nothing at all. The same is true *mutatis mutandis* of 'if *P*, then *P*', '*P* and not *P*', 'if *P*, then *P* or *Q*', and so on. *If* those shapes are taken to have certain meanings, *then* there is a function—embodied in the method of truth tables—assigning truth or falsity to certain expressions depending on their shapes. So using truth tables to perform a computation (to determine truth-values) thus presupposes semantic background knowledge; it presupposes knowledge that certain semantic conventions are operative.

[27] See Coulter (1983, pp. 57–58) and Wittgenstein (1958; 1983, p. 422, especially: "'Mechanical'—that means: without thinking. But *entirely* without thinking? Without *reflecting*").

[28] If you are a talented pianist, you can automatically play (or even sight-read) a piece. If you are an extremely talented composer, you can automatically compose a sonatina. But these operations though "automatic" in some sense, are not "mechanical"; such operations though executed automatically obviously implicate a rich backlog of knowledge and of cognitive abilities.

[29] Again, it may be appropriate to reference Wittgenstein (1983, especially pp. 257–258, 422).

[30] At least not in virtue of the fact that it is manipulating shapes: there is always the possibility that, for some entirely unrelated reason, it is engaging in cognitive processes.

[31] Suppose it turns out that thought *does* consist in morphology-driven operations on symbols. This is, I suppose, an epistemic possibility. Brain research might reveal that a person is thinking *Socrates is bald, therefore something is bald* exactly if there occurs in that person a sequence of brain states with certain morphologies. (This is not my own belief; but I do not wish to deny, at least not in this context, that this *might* turn out to be the case.) But if it turned out that thought consisted in purely morphology-driven sequences of brain states (a very dubious empirical proposition), that would not (at least not in any direct way) vindicate the idea that thought is fundamentally *computational*. For a computation is not a manipulation of symbols; a computation is *not* a morphology-driven sequence of symbols. A computation is a *judgment* regarding the validity of strings of symbols. Such a judgement may, or may not, result in a manipulation of shapes.

References

Barwise, J., & Perry, J. (1999). *Situations and attitudes*. Palo Alto, CA: CSLI Press.

Burge, T. (1979). Individualism and the mental. In P. A. French, T. E. Uehling, & H. K. Wettstein (Eds.), *Midwest studies in philosophy: Vol. 4. Studies in metaphysics* (pp. 73–121). Minneapolis, MN: University of Minnesota Press.

Cain, M. J. (2002). *Fodor: Language, mind, and philosophy*. Malden, MA: Blackwell.

Carnap, R. (1947). *Meaning and necessity*. Chicago: University of Chicago Press.

Chomsky, N. (1975). *Reflections on language*. New York: Pantheon Books.

Coulter, J. (1983). *Rethinking cognitive theory*. New York: St. Martin's Press.

Fodor, J. (1975). *The language of thought*. New York: Thomas Y. Crowell.

Fodor, J. (1981). *Representations*. Cambridge, MA: MIT Press.

Fodor, J. (1987). *Psychosemantics*. Cambridge, MA: MIT Press.

Fodor, J. (1990). Fodor's guide to mental representation. In *A theory of content and other essays*. Cambridge, MA: MIT Press.

Fodor, J. (1994). Fodor. In S. Guttenplan (Ed.), *A companion to the philosophy of mind*. Cambridge, England: Blackwell.

Fodor, J. (1998). *Concepts*. Oxford, England: Clarendon Press.

Fodor, J. (2001). *The mind doesn't work that way*. Cambridge, MA: MIT Press.

Frege, G. (1892). Über Begriff und Gegenstand. In *Vierteljahresschrift für wissenschaftliche Philosophie*, *16*, 192–205.

Kaplan, D. (1979). On the logic of demonstratives. *Journal of Philosophical Logic*, *8*, 81–98.

Kaplan, D. (1989). Demonstratives. In J. Almog, J. Perry, & H. Wettstein (Eds.), *Themes from Kaplan* (pp. 481–566). Oxford, England: Oxford University Press.

McCawley, J. D. (1981). *Everything that linguists have always wanted to know about logic but were ashamed to ask*. Chicago: University of Chicago Press.

Montague, R (1973). The proper treatment of quantification in ordinary English. In J. Hintikka (Ed.), *Approaches to natural language* (pp. 221–242). Dordrecht, Netherlands: Reidel.

Putnam, H. (1975). The meaning of 'meaning'. In *Philosophical papers: Vol. 2. Mind, language, and reality* (pp. 215–271). Cambridge, England: Cambridge University Press.

Pylyshyn, Z. W. (1984). *Computation and cognition*. Cambridge, MA: MIT Press.

Quine, W. V. O. (1966). Quantifiers and propositional attitudes. In *The ways of paradox* (pp. 185–196). Cambridge: Harvard University Press.

Quine, W. V. O. (1970). *Philosophy of logic*. Cambridge, MA: Harvard University Press.

Russell, B. (1905). On denoting. *Mind, 14,* 479–493.

Strawson, P. F. (1950). On referring. *Mind*, *59*, 320–344.

Wittgenstein, L. (1958). *The philosophical investigations*. Oxford, England: Basil Blackwell.

Wittgenstein, L. (1983). In G. H. von Wright, R. Rhees, & G. E. M. Anscombe (Eds.), *Remarks on the foundations of mathematics* (pp. 5–437). Cambridge, MA: MIT Press.