To appear in *The Symbolic and Connectionist Paradigms: Closing the Gap* (J Dinsmore, ed.). Lawrence Erlbaum, 1992.

Subsymbolic Computation and the Chinese Room

David J. Chalmers

Center for Research on Concepts and Cognition
Indiana University
Bloomington, Indiana 47405
E-mail: dave@cogsci.indiana.edu

1 Introduction

More than a decade ago, philosopher John Searle started a long-running controversy with his paper "Minds, Brains, and Programs" (Searle, 1980a), an attack on the ambitious claims of artificial intelligence (AI). With his now famous *Chinese Room* argument, Searle claimed to show that despite the best efforts of AI researchers, a computer could never recreate such vital properties of human mentality as intentionality, subjectivity, and understanding.

The AI research program is based on the underlying assumption that all important aspects of human cognition may in principle be captured in a computational model. This assumption stems from the belief that beyond a certain level, implementational details are irrelevant to cognition. According to this belief, neurons, and biological wetware in general, have no preferred status as the substrate for a mind. As it happens, the best examples of minds we have at present have arisen from a carbon-based substrate, but this is due to constraints of evolution and possibly historical accidents, rather than to an absolute metaphysical necessity.

As a result of this belief, many cognitive scientists have chosen to focus not on the biological substrate of the mind, but instead on the abstract causal structure that the mind embodies (at an appropriate level of abstraction). The view that it is abstract causal structure that is essential to mentality has been an implicit assumption of the AI research program since Turing (1950), but was first articulated explicitly, in various forms, by Putnam (1960), Armstrong (1970) and Lewis (1970), and has become known as *functionalism*.

From here, it is a very short step to *computationalism*, the view that computational structure is what is important in capturing the essence of mentality. This step follows from a belief that any abstract causal structure can be captured computationally: a belief made plausible by the Church–Turing Thesis, which articulates the power of simple computational structure. Some, nevertheless, have chosen not to make this step. Penrose (1989), for instance, seems to be a functionalist without being a computationalist, holding that certain causal properties of the mind cannot be captured algorithmically. Since Putnam's Turing–machine model of mental states, however, computationalism has been the most popular form of functionalism. In the eyes of many, the terms “functionalism” and “computationalism” are synonymous.

Searle’s term for the doctrine of computationalism is *"strong AI”*. This contrasts with *weak AI*, which holds merely that the computer can be used as a powerful tool for understanding the mind. Strong AI goes far beyond this, holding that the appropriately programmed computer really *has* (or *is)* a mind. In Searle’s view, strong AI is thoroughly implausible, and his arguments attempt to discredit it, showing that nothing could ever possess true mentality merely by virtue of instantiating an appropriate computer program.

On the issue of computationalism, the artificial intelligence research community is unsurprisingly in near-universal agreement: there exists a certain class of computer programs (none yet written), such that anything that instantiates one of these programs can be correctly said to have (or be) a mind. There is nevertheless significant disagreement over just what this class of programs is. In recent years, artificial intelligence researchers have split into two polarized camps.

One camp, the *symbolic AI* school, has the weight of 3 decades’ tradition behind it. The symbolic camp maintains that the correct level at which to model the mind is that of the *symbol*—that is, an entity in a computer program that is taken to refer to some entity in the real world. Programs that instantiate mentality, this camp claims, will be a subset of the class of programs that perform computation directly on such symbols. This view has been carefully articulated by Newell and Simon (1976), and is discussed in more detail later.

The *connectionist* school, by contrast, has only recently become a serious contender, although it has grown out of less visible research from the last few decades. On the connectionist view (Rumelhart, McClelland, & the PDP Research Group, 1986), the level of the symbol is too high to lead to a good model of the mind. We have to go lower: instead of designing programs that perform computations on such symbols, design programs that perform computations at a lower level. The correct level is sometimes taken to be the level of the neuron, as the basic unit of many connectionist models has a causal structure reminiscent of the neuron, although Smolensky (1988) cautions us against making too much of this similarity. It may be that going to the level of the neuron is going deeper than necessary. Some connectionist models, and certainly the related models of Holland (1986) and Mitchell and Hofstadter (1990) are better taken as working at the *subsymbolic* level. This is a level below that of the symbol—unlike symbolic models, these models have no direct correspondence between the entities computed upon and entities in the world—but not necessarily as deep a level as that of the neuron. These systems still follow rules, but the rules are well below the semantic level. It is hoped that as a consequence of following rules at this low level, semantic properties will emerge—that is, manifest themselves in the processing and behavior of the program—without having been explicitly programmed in. Consequently, when viewed at the semantic level such systems often do not appear to be engaged in rule-following behavior, as the rules that govern these systems lie at a deeper level.

Searle views each of these enterprises as misguided. Both hold that some computational structure is sufficient for mentality, and both are therefore futile. Searle's position is that his arguments show that neither of these schools of research could produce a truly cognitive system. In this chapter, I examine Searle's arguments carefully to see how they apply to both schools of artificial intelligence research. We find that Searle's arguments provide at least a plausible criticism of the symbolic AI research program, but that connectionist models are less vulnerable to his arguments.

In fact, it turns out that some of Searle's arguments are almost equivalent to those put forward by some members of the subsymbolic school as criticisms of their more traditional colleagues. It seems that perhaps Searle should desist from his blanket condemnation of the entire AI research endeavor, and instead sign up for his connectionist calling-card as soon as possible. This seems a little unlikely, however.

2 Searle's arguments

2.1 The Chinese Room

Searle first propounded his arguments against computationalism in terms of the Chinese Room thought-experiment. The argument, briefly, goes something like this: Assume, along with AI researchers, that it is possible to write a computer program that is an accurate model of the mind. In particular, such a program would then be an accurate model of human behavior, and would be able to pass the Turing Test, producing verbal behavior indistinguishable from that of a human. For added color, Searle asks us to imagine a program that produces fluent linguistic behavior in Chinese.

Proponents of strong AI claim that such a program, when implemented, would produce all the essential properties of mentality, including intentionality, consciousness, and so on. In particular, any implementation of such a program could fairly be said to understand Chinese in the same way that a human speaker of the language would. This is the claim with which Searle takes issue.

To convince us of this, Searle asks us to imagine a run of the program being simulated by a human who speaks no Chinese. The instructions in the program are formulated as a set of rules (in English) for the human to follow, stepping through the program by hand. The human is isolated in a secure room, and batches of Chinese writing are passed into the room. The human understands no Chinese, but using the supplied rules (representing the program), undertakes a series of manipulations involving symbols on paper, yielding as an end-product certain other shapes on paper, which are in fact Chinese symbols that represent the output of the program. Those outside the room are unaware of the internal details—they see only a room which is somehow passing a Chinese Turing Test.

It is clear, Searle tells us, that the human does not understand Chinese. To the human, the manipulated symbols are simply meaningless squiggles and squoggles. This is intuitively clear, and few have disputed this point. From here, Searle concludes that a computer running the same program would understand Chinese no better. In this way, strong AI is refuted.

Many problems have been pointed out with Searle's argument. The most common reply has been to note that whereas few would suggest that the human understands Chinese, it is not implausible that the system as a whole does. Searle dismisses this idea, finding it ridiculous that a system of pens, paper and human could ever understand anything, but this may be a limit of Searle's intuition. (It has been pointed out that in practice, such a Chinese Room would almost certainly be tortuously complex, with one input-output sequence taking days, months or years on the part of the human. Faced with such complexity, our intuitions tend to break down.) Searle defends the argument against this reply, the so-called *Systems Reply*, but many have found his defense unconvincing. It seems to many that Searle is committing the mistake of confusing two different systems that have a common location.

2.2 Syntax and Semantics

Faced with such criticisms, Searle streamlined his argument to its essence, as follows (Searle, 1984, p. 39; Searle, 1987, pp. 231–232):

Axiom 1: Syntax is not sufficient for semantics.

Axiom 2: Minds have contents; specifically, they have semantic contents.

Axiom 3: Computer programs are entirely defined by their formal, or syntactical, structure.

Conclusion: Instantiating a program by itself is never sufficient for having a mind.

This is the form of the argument on which this chapter focuses. The original Chinese Room thought-experiment is a rich source of intuitions, and it is not impossible that some of the force of the original argument might be lost in this bare-bones reformulation. (I discuss some other aspects of the Chinese Room argument in Chalmers, 1991.) Nevertheless, the syntax/semantics argument is Searle's preferred formulation, and it seems to capture a significant intuition behind the original thought-experiment. It is toward this argument, and toward the corresponding intuition, that my remarks are addressed.

Each of the axioms above is superficially plausible. To determine just how compelling the argument is, however, we will need to examine exactly what is meant by "syntax" and "semantics," and see how the truth of the axioms depend on the definitions of these terms.

The key axiom here is the first: "Syntax is not sufficient for semantics." This seems plausible indeed: Have we not all learned from our explorations in folk linguistics that syntax and semantics are very different things? It is probably safe to assume that Searle's motivation for this axiom comes directly from linguistics. It says, effectively, that the way in which objects (words in language, computational entities in artificial intelligence) obey formal rules is independent of the way in which they have meaning.[1]

This axiom, it seems, represents for Searle a distillation of the main thrust of the Chinese room argument. When the human in the room is manipulating symbols on paper, all she sees are "meaningless squiggles and squoggles." Although she is able to follow the (syntactic) rules perfectly, she has no idea what the symbols *mean*, and so she cannot be said to have any true understanding of the Chinese language.

The way this argument can be used against computationalism is clear. Computer programs may manipulate symbols labeled **RESTAURANT**, **ELEPHANT**, and so on, but they will still have no idea what a restaurant is, and they will not understand the concept "elephant". The entities manipulated by the computer, in Searle's view, are hollow, empty symbols, devoid of meaning. The symbols, and therefore the program, possess no intrinsic semantics.

Before proceeding any further, it is important to clarify precisely to which brand of semantics this argument applies. Meaning comes in a variety of guises. One sort of meaning is externalist, *extensional* semantics. Under this construal, something possesses meaning only if it picks out some definite object or set of circumstances in the external world. Here, meaning is essentially a world-involving notion. A symbol or mental state has meaning only in virtue of a relation to some external object—an *extension*—and such an extension can only be acquired by appropriate causal interactions with the external world.

Searle's notion of semantics is not, however, an externalist notion. When Searle claims (in Axiom 2) that "Minds possess semantic contents," he is referring to *internalist* semantic contents. These are semantic contents that the mental states possess independent of specific environmental details. Roughly, they are *intensions* rather than extensions. Searle (1980b) makes this internalism very clear, arguing that even a disembodied brain possesses semantic

[1] More accurately, the rule-following behavior does not *determine* the meaning. Recent work in linguistics (e.g. Lakoff, 1987) has demonstrated that syntax and semantics are not completely independent.

content:

> If I were a brain in a vat I could have exactly the same mental states I have now; it is just that most of them would be false or otherwise unsatisfied. Now this [...] is designed to make clear what I have in mind when I say that the operation of the brain is causally sufficient for intentionality, and that it is the operation of the brain and not the impact of the outside world that matters for the content of our intentional states, at least in one important sense of “content”. (p. 452)

Searle’s notion of content is what is referred to in the philosophical literature as a *narrow* notion of content: Content that is possessed by an individual merely in virtue of physical properties of that individual, and not in virtue of properties of the environment. This contrasts with *wide*, or environment-involving content.[2] In fact, Searle’s notion of content is sufficiently internalist that according to this notion mental states only count as truly semantic if they are conscious or potentially conscious, as a recent article (Searle, 1990) makes clear.[3]

This internalist/externalist distinction serves to cut off a potential escape route from the syntax/semantics argument. If externalist semantics were under discussion, one might argue that certainly purely syntactic computation, operating in a vacuum, cannot give rise to semantics; but if we simply hook up the computer to the real world in the right way, then its states will acquire semantic content in virtue of their causal interaction with the world, just as human mental states acquire extensional content in virtue of causal interaction with the world. This escape route, which Searle dubs the *Robot Reply*, will not work for internalist content, as this kind of content is independent of environmental factors. If a system cannot possess internalist content on its own, environmental hook-up will not help.

The thrust of Searle’s argument, then, is that computers can manipulate symbols, but there is nothing internal to the system that gives the symbols any meaning. If this is true, then computers stand in clear contrast to humans, whose mental states possess intrinsic meaning.

[2] The literature on narrow and wide content started with Putnam (1975), and has been developed by Fodor (1980) and Burge (1984), among many others.

[3] It should be noted that an internalist view on content does not preclude content from being a *referential* notion. A mental state may possess internalist semantic content by virtue of reference to some object or concept in a subject' s *notional world* (Dennett, 1982), rather than by virtue of reference to the external world (as is the case for externalist content). A subject' s notional world will usually match up quite well with the external world, but need not, as evidenced by the case of the brain in the vat.

3 Symbolic and Connectionist AI

3.1 Symbolic AI

Before we can determine how well Searle's arguments apply to the symbolic and connectionist artificial intelligence endeavors, we must outline and compare the fundamental assumptions and commitments of these two schools. We start with the symbolic AI, whose underlying assumptions have been well stated by Newell and Simon, two of the founders of the field.

Newell and Simon (1976) explicitly state the assumptions on which the symbolic AI research program rests, and group these under the name of the "Physical Symbol System Hypothesis." This states, simply:

> A physical symbol system has the necessary and sufficient means for general intelligent action.

To Newell and Simon, an entity is potentially intelligent if and only if it instantiates a physical symbol system: a system, embodied physically, that is engaged in the manipulation of symbols.

The most important property of a symbol, to Newell and Simon, is that it *designate*. A symbol is not a symbol unless it symbolizes something. Further, symbols are *atomic*. Symbols may combine together to form expressions, but they may not be broken down. The most obvious example of such symbols comes in a programming language such as LISP. Symbols, such as **CLYDE**, **ELEPHANT** and **TREE** are implemented as LISP atoms. These refer to objects or concepts, in these cases those of *Clyde*, *elephant*, and *tree* respectively. The symbols may be combined to form complex structured expressions, which refer to more complex entities. A complex entity may be represented to by a property list, for instance: as a simplistic example, a circus might be represented by **(CIRCUS (NUMBER-OF-PERFORMERS 23) (ELEPHANT CLYDE) (LOCATION BIG-TOP))**. There is some room for debate, within the symbolic tradition, over just which mental representations are symbols and which are expressions. For instance: *bachelor* is presumably a compound concept (formed out of lower level expressions such as *unmarried* and *male*), but is *elephant*? Might this be an expression rather than an atomic symbol, formed out of basic concepts such as *gray*, *animal*, and *trunk*? Whatever the correct level for atomic symbols, however, no question is raised that this level is a semantic one. Atomic symbols must carry a representational burden.

A symbol, then, is an atomic entity, designating some object or concept, which can be manipulated explicitly by a physical symbol system, leading to intelligent behavior. Symbolic AI deals with the class of programs that perform computations directly upon such symbols.

3.2 Connectionist AI

Connectionism, by contrast, eschews atomic symbols. Connectionism is usually taken not to be symbolic, but subsymbolic. The best account of the assumptions underlying connectionism has been given by Smolensky (1988). Smolensky offers the following *Subsymbolic Hypothesis* as "the cornerstone of the subsymbolic paradigm":

> The intuitive processor is a subconceptual connectionist dynamical system that does not admit a complete, formal, and precise conceptual level description. (p. 7)

The second clause here is important, for it is what establishes connectionist and symbolic AI as distinct endeavors. The Physical Symbol System Hypothesis holds that all mental processing consists in computation on atomic symbols. If this hypothesis is true, then there exists a complete description of mental processing at the level of the atomic symbol. A description in these terms will necessarily be a description at the conceptual level, as atomic symbols are by definition representations of concepts. The Subsymbolic Hypothesis explicitly holds that no such description exists. Rather, to give a full account of mental processes, one must invoke processes that lie beneath the conceptual level.

There is certainly computation in connectionist systems, but such computation lies below the conceptual level. Computation takes place at the level of nodes and connections between nodes, and the individual nodes and connections are not intended to carry any semantic burden alone. The semantic burden of the system lies at a higher level, that of the *distributed representation* (Hinton, McClelland, and Rumelhart, 1986). Such a representation consists of a pattern of activity over a number of different nodes. In virtue of this distributed pattern, the representation has a complex internal structure that plays an important causal role. A distributed representation, then, is certainly not atomic. Neither is it a complex *expression* in the symbolic style, made up of simpler atomic representations. The components of the representation—the individual nodes or connections—carry no semantic burden and are therefore not representations in their own right. Connectionist systems therefore avoid the use of atomic symbols, and the connectionist endeavor consequently rejects the Physical Symbol System Hypothesis.

3.3 Connectionist and Symbolic Systems: A Comparison

It is important to clarify precisely how the symbolic and connectionist endeavors differ. Some have tried to isolate the differences between symbolic and connectionist systems purely in virtue of their architecture. The connectionist endeavor might be distinguished solely by its reliance on neuromorphic processing, with neuron-like units and synapse-like connections between units, whereas the symbolic endeavor is identified by its reliance on von-Neumann-style processing, or by architectures based on the Turing machine, or by the use of programs formulated in programming languages. Such a distinction is unlikely to be useful, however. For a start, both von Neumann and the neural-network architectures are *universal*: Anything that one can do, the other can do also (Franklin and Garzon, 1990). Indeed, connectionist models are typically formulated in high-level programming languages, and implemented on von Neumann machines, but this does not make them any less connectionist.

Adams *et al* (Chapter 1 in this volume) compare the capacities of programming-language and neural-network architectures, and find no significant differences. This should not be too surprising. Both kinds of architecture should be regarded as tools on which various kinds of cognitive models can be implemented. Even within the symbolic school, high-level programs on von Neumann machines have for years been used as implementation bases on which other virtual architectures—such as production systems, script-based memories and so on—have been overlaid. The situation with neural networks is no different. Like production systems, connectionist models can be implemented equally well on a Turing-machine or a neural-network base. Connectionist models no more imply a rejection of the Turing machine model of computation than do production systems or blackboard architectures.

It follows that the connectionist and symbolic schools should not be distinguished by *syntactic* criteria alone. The styles of formal computation used may appear identical. Rather, the relevant criteria are *semantic*. It is the way in which representations are used that is fundamental. To be precise, the schools are distinguished by the way in which their computational (syntactic) features relate to their representational (semantic) features. To see this, consider the role of syntax and semantics within symbolic and connectionist systems.

Consider syntax first. In any computational system, the fundamental syntactic entities are *computational tokens*: These are the atomic objects that are manipulated in order for the computation to take place. In a neural network, the computational tokens—the elementary indivisible units—are individual nodes and connections. In a typical symbolic system, the computational tokens might be LISP atoms.

The fundamental semantic entities of any system, on the other hand, are *representations*: By definition, these are the objects that carry the semantic burden of the system. In a connectionist system, a representation is typically a distributed pattern of activation over a set of nodes. In a symbolic system, a representation might be a LISP atom or expression.

Both connectionist and symbolic systems, then, possess both computational tokens and representations, so the two classes cannot be distinguished by the possession of either of these features alone. They can, however, be distinguished by the relationship they suppose between the two. In symbolic systems, representations and computational tokens coincide. Every basic representation is an atomic computational token, and other representations are built by compounding basic representations. A LISP atom, for instance, serves simultaneously as a computational token and as a representational vehicle. This is precisely the import of the Physical Symbol System Hypothesis. A *symbol*, in Newell and Simon's usage, is an object in a computational system that (a) is atomic, that is, is a computational token, and (b) designates, that is, is a representation.

In connectionist systems, by contrast, representations and computational tokens are quite separate. Individual nodes and connections, the computational tokens, lie at a completely different level of organization from representations, which are distributed patterns of activity over such tokens. This is the most important distinguishing feature of connectionist systems: the level of computation falls below the level of representation. This distinction, philosophically, is much more important than the choice of any particular architecture. Some connectionists have proposed that this feature be the defining feature of connectionism, but for many the term "connectionism" is too closely bound to the neural-network architecture. Hence the coining of the term "subsymbolic". A subsymbolic system is precisely one in which the level of computation falls below the level of representation.[4]

We may enshrine this distinction into the two distinct philosophical commitments of the symbolic and subsymbolic schools. These commitments define symbolic and subsymbolic cognitive models.

> *The Symbolic/Subsymbolic Distinction:* In a symbolic system, the computational level coincides with the representational level. In a subsymbolic system, the computational level lies beneath the representational level.

There are a couple of points worth clarifying. First, we do not want to assume in advance that either kind of system possesses intrinsic semantics, in Searle's sense. So at least for now, talk of representations should be regarded as *interpretational* talk. An object within a computational system is a representation if it is an object of our semantic interpretation. Perhaps such an object of interpretation possess honest-to-goodness intrinsic semantics, and perhaps it does not. That remains to be seen. In these terms, another way of phrasing the relevant distinction is as follows:

[4] The term "subsymbolic" is perhaps best viewed as modifying the noun "computation", so that a "subsymbolic system" is shorthand for "a system utilizing subsymbolic computation". The computation is subsymbolic in virtue of falling below the representational level. Perhaps a better term would be "subrepresentational", due to the many connotations of the word "symbol", but the term "subsymbolic" seems to be around for the long haul.

> *The Symbolic/Subsymbolic Distinction* (alternative version): In a symbolic system, the objects of computation are also objects of semantic interpretation. In a subsymbolic system, the objects of computation are more fine-grained than the objects of semantic interpretation.

Second, the symbolic/subsymbolic distinction completely cross-classifies the architectural distinction between Turing machines, say, and neural networks. Turing machines can be used to implement both symbolic and subsymbolic models, as can neural networks. Indeed, the well-known *localist* neural-network models, in which concepts are represented by single nodes, have representations that coincide with computational tokens. Such localist models are therefore better regarded as symbolic than subsymbolic, despite their use of a neural-network architecture. On the other hand *distributed* neural-network models, which lie at the core of the connectionist endeavor, are paradigm examples of subsymbolic computation. In future, when I use the term "connectionist" alone, I will be referring to distributed, and therefore subsymbolic, connectionist models.

Finally, some might argue that even traditional symbolic models can be construed as engaging in "subsymbolic computation". In a typical symbolic system, a basic representational token—a LISP atom, say—might be implemented as a pattern of bits within the circuitry of the computer. One might argue that the real computation is going on at the bit level, and not at the level of this LISP atom. If this was the case, then the level of computation would fall below the level of representation, and we would have subsymbolic computation on our hands.

In response to this argument, it should be pointed out that whereas such bits might be entering into a computation, they are not individually causally efficacious. Whenever they enter a computation, they enter as a chunk—for instance, the eight bits that implement the atom **ELEPHANT** are always manipulated together. For all intents and purposes, these eight bits form an atomic unit. They stand in a strictly *implementational* relationship to the computation at LISP-atom level, and so do not have any significance of their own. The eight bits merely form an arbitrary label for the LISP atom; any collection of bits would do as well. This contrasts clearly with the connectionist case. In a connectionist network, a representation is made up of a collection of activated nodes, but each of these nodes plays a separate and significant causal role. We could not replace this particular configuration of active nodes by another configuration without radically altering the causal role of the representation. The distributed representation, unlike the LISP atom, cannot be treated as an atomic object.

The moral here is that the computation at stake in distinguishing symbolic from subsymbolic computation must be causally efficacious. If a computation is merely implementing a separate algorithm at a higher level of description, then we should move to that higher level. For a model to be truly subsymbolic, it must not only be the case that it engages in subsymbolic computation: It must also be the case that it cannot be redescribed as engaging in symbolic computation at a higher level. Otherwise the subsymbolic computation of the model would be a merely implementational, causally irrelevant feature. This coheres well with Smolensky's Subsymbolic Hypothesis, which states that mental processes form a subconceptual dynamical system, and cannot be completely described at the conceptual level. This hypothesis, translated into the language that we have been using, states that the mind engages in truly subsymbolic computation, and therefore cannot be redescribed as engaging in symbolic computation.

4 Searle's argument and symbolic/subsymbolic computation

Now that we have our distinctions clear, we may apply them to Searle's argument, and see how successfully the argument applies to the two distinct brands of AI. We have seen that the symbolic and subsymbolic schools take very different views of the relationship between syntax and semantics, so it would not be surprising if Searle's argument applied differently in each case.

Before proceeding, there is one overly simple view of the application of Searle's argument to connectionism that we can readily dismiss. This view holds that the argument should not be construed as even attempting to apply to connectionism; rather, the argument is a critique solely of symbol-manipulating systems. This argument is fallacious: Searle's argument is aimed at computational systems in general, and connectionist systems are computational systems. Indeed, almost all connectionist models are implemented as programs, so the argument applies directly.

The fallacy in this argument trades on an ambiguity in the term "symbol". Sometimes, the term "symbol" is taken to be synonymous with "computational token"—as, for example, when one regards a one or zero in a Turing Machine as a symbol irrespective of whether it carries any representational content. In this sense, Searle's argument indeed is aimed at only symbol-manipulating systems—that is,. systems that manipulate computational tokens—but this is not much of a restriction. More frequently, however, the word "symbol" is used to mean something intended to possess representational content—most typically, a computational token that also represents, as we saw above. In this sense, the only symbol-manipulating programs are those of symbolic AI. This does not mean that Searle's argument is directed only at symbolic AI, however, as the term "symbol-manipulating" is being used differently in different parts of the argument.

4.1 Searle and Symbolic AI

Symbolic models, as we have seen are distinguished by their manipulation of atomic symbols—primitive computational tokens that are intended to carry semantic content. This leaves them vulnerable to Searle's argument in a very direct way. In virtue of what do such tokens posses intrinsic content?

First, *interpretational* content alone is not enough. To be sure, we may interpret such tokens as possessing content, but the possibility of such interpretation cannot guarantee intrinsic semantics, as witnessed by the case of words on a page. The word "dog", written on a page, is interpreted by an observer as referring to the concept of dog, but such a word surely does not possess intrinsic semantic content. Rather, its semantic content is derived entirely from the semantic content of the observer, and is therefore merely *extrinsic* content.

Furthermore, we cannot argue that such tokens accrue content by virtue of a causal relationship to an object in the external world. One might wish to argue that the token "dog" represents the concept *dog* in virtue of the fact that the token is activated by the presence of real dogs, but such an argument would only work if it was externalist content that was at stake. As we saw earlier, however, Searle's argument is directed at internalist content, and so environmental properties are irrelevant.

We therefore need to find some intrinsic property by virtue of which a computational token can possess content. But here, we come up against the basic problem: these tokens are *atomic*. They possess no relevant internal structure. A token representing the concept *elephant* may be labeled **ELEPHANT**, but it might equally be labeled **APPLE** for all the program cares. Internal to the computation, such a token might consist of some pattern of bits, but any pattern of bits will do equally well. For all intents and purposes, a computational token is a featureless chunk, coming only with an arbitrary label that only serves to distinguish it from other computational tokens. Nothing intrinsic to the **ELEPHANT** token makes it any more closely related to the *elephant* concept than to the *apple* concept.

This seems to be precisely the import of the intuition behind Searle's argument. As far as system is concerned, the token "elephant" is meaningless. To the computer, such a token no more carries semantic content than do Chinese symbols to the English-speaker in the Chinese Room. It is a featureless object that is shoved around within the system, with nothing, apart from an arbitrary label, to distinguish it from other tokens within the system. As a syntactic primitive, there is nothing about it that qualifies it for meaning.

Searle's argument that "syntax is not sufficient for semantics" is effectively claiming that no amount of syntactic manipulation of such an object can endow it with true meaning. I should make clear that I am not necessarily endorsing this argument. One possible reply, for instance, might be that such a token does not represent in virtue of any intrinsic properties, but instead in virtue of its relationship to other tokens. Searle would argue that such relational properties cannot help—that an extrinsic relation between one primitive token and another cannot endow either token with meaning. However, the concern here is not to evaluate the ultimate success of this argument. Instead, I am pointing out that symbolic systems may be vulnerable to the argument in a way that subsymbolic systems may not be. Searle's argument derives its force from a set of powerful intuitions; I am trying to isolate these intuitions, and to see how they apply in various cases. The chief intuition behind Searle's argument appears to be that manipulation of tokens cannot endow these tokens with meaning. If this intuition is correct, then it certainly applies strongly to symbolic models.

4.2 Searle and Subsymbolic AI

In contrast to symbolic AI, connectionism eschews the usage of atomic symbols altogether. In connectionist models, and in other subsymbolic models, the level at which rule-following manipulation occurs is not a semantic level at all, and makes no claims to be. The fundamental idea behind connectionism, as we have seen, is to engage in algorithmic processes at a lower, nonsemantic level. The hope is that a semantic level will be emergent from this level, not unlike the way in which the laws of thermodynamics are emergent from the motion of individual particles.

One key property of connectionist systems, as we saw earlier, is that in these the meaning of the term "symbol" has come apart into two pieces. In such systems there are representations, and there are computational tokens, but the two classes are quite separate. This separation goes a long way toward safeguarding these systems from Searle's argument. To see this, let us try to run the argument through against these models. The argument, recall, is that symbol manipulation cannot give these symbols meaning. As the term "symbol" has now split into two, we may concentrate on either of the two relevant concepts: computational tokens or representations.

First, consider computational tokens, such as nodes and connections. It is true that these are primitive, featureless entities in connectionist systems, with nothing to distinguish them except perhaps a numerical degree of activation. Therefore, running Searle's argument through, such tokens cannot possess any intrinsic semantic content. But this is no problem for the connectionist! Such tokens were never intended to possess content; unlike their counterparts in symbolic models, these tokens are only intended to be syntactic objects. As is made clear by the symbolic/subsymbolic distinction, the computational level in connectionist systems is not a representational level. Therefore Searle's argument, applied to computational tokens, tells us nothing that we did not already know.

Second, let us consider representations. Representations in connectionist systems, unlike those in symbolic systems, are not primitive, featureless entities. Rather, a connectionist representation is a complex distributed pattern of activity, emerging from computational activity at a lower level. Representations are not manipulated directly, but rather are the indirect result of low-level manipulations. Most importantly, connectionist representations have a rich *internal structure*. Unlike symbolic representations, connectionist representations have their own intrinsic organization, by virtue of their consisting in a complex pattern of activation; it is impossible to treat such a representation as a featureless chunk. Because of this internal structure, Searle's argument does not go through. According to the connectionist claim, distributed representations carry content precisely in virtue of their rich internal structure. In the terms of Dyer (1990), connectionist representations have a *microsemantics*—nternal pattern that systematically reflects the meaning of the representation.

The internal structures of symbolic representations, we saw, are entirely interchangeable. Any token will serve as well as any other, and there is nothing about the label **ELEPHANT** that makes it serve any better as a label for the atomic representation of the *elephant* concept than as a label for the *dog* concept. By contrast, the internal structure of a connectionist representation is of central importance. If we take the representations of two concepts—say *elephant* and *dog*—and attempt to exchange the relevant patterns of activation, the system will not function properly. If we attempt to modify the system of weights from these representations to compensate for this exchange, all the other representations in the system will be interfered with. The internal structure of a connectionist representation is essential to the system's function, and it is not the case that any old pattern of activity will do. The specific pattern of activity within a representation is responsible for making it represent what it does.

We can therefore see that Searle's intuition is not nearly as applicable in the connectionist case as in the symbolic case. There are both syntactic objects (computational tokens) and semantic objects (representations) in a connectionist system, but the syntactic objects are not semantic objects and vice versa. In this way the syntax/semantics argument is avoided.

4.3 Is Syntax Sufficient forSemantics?

Searle would doubtless be unimpressed by the argument above. He might point to his axioms and say: (1) connectionist models are computational, consisting in syntactic manipulation of computational entities; (2) syntax is never sufficient for semantics; therefore (3) connectionist models can possess no true semantics. Even though distributed representations are not primitive tokens, they result entirely from syntactic manipulations, and so can carry no true semantic content.

We must therefore consider whether it is possible for syntactic systems to possess semantic content at any level. First, what might it mean for a system to be syntactic? Presumably, this means that the system functions by following rules, whether at a low level, such as the level of the node and connection in a connectionist system, or at a high level, such as the level of the representation in a symbolic model. Second, what does it mean here for a system to possess semantic content? As we have seen, the notion of content at stake here is internalist, intensional content. Putting these together, we may translate the statement about syntax and semantics approximately as follows: "No system that consists entirely of rule-following behavior can possess internal content."

Unfortunately, this axiom as it stands is false, as we may see by considering the example of the human brain. The brain surely possesses internal content, if any system does. Yet the brain can in principle be described as following a set of iron-clad rules: namely, the laws of physics. On the highest level, the human mind seems extremely flexible, producing the very antithesis of rule-following behavior; yet at the bottom level, it is made up of a physical substrate, consisting of such entities as elementary particles and electrical charges, whose actions are determined by the laws of physics. Of course, in our current state of knowledge we do not have complete knowledge of these laws, but it is a tenet of modern science that a complete set of such laws exists. At the lowest level, neural processing is not sensitive to any semantic properties; rather, each neuron fires or not depending only on content-free properties of input signals from other neurons. At some level, then, the brain is a rule-following, syntactic device. Nevertheless, it certainly possesses internal content, so Searle's "axiom" is false.

Consequently, it is simply not true that no syntactic system can possess content. Nevertheless, we have seen that there is some strong intuition behind the claim that "Syntax is not sufficient for semantics" that might be worth saving. If we modify the force of this axiom appropriately, we might be able to arrive at a formulation that applies to the cases where the intuition is most compelling, while at the same time excluding the obvious counterexamples such as the human brain.

Where does the motivation for the "Syntax is not sufficient for semantics" axiom come from? It seems clear that it comes from linguistics: The syntactic rules that words obey are not enough, alone, to endow these words with semantic content. For a *sentence*, then, it is true that syntax is not sufficient for semantics. We must therefore find some relevant difference between brains and sentences in virtue of which syntax is sufficient for semantics in one case but not in the other. A plausible answer immediately suggests itself: While it is true that brains are syntactic, their syntax lies at an extremely low level. The syntactic properties of atoms, molecules, even neurons seem almost irrelevant when we are talking about the conceptual level. Such syntax lies so far down that it does not diminish any semantic properties of *concepts*. In the case of the sentence, by contrast, the syntax and the intended semantics lie at precisely the same level, the level of the word. It is words that are manipulated syntactically, and it is words that are the object of semantic interpretation.

We might therefore patch up Searle's axiom as follows. Instead of the blanket pronouncement "Syntax is never sufficient for semantics," we might instead make the more limited conjecture: "Syntax, at a certain level, is never sufficient for semantic content at the same level." This conjecture comes much closer to capturing our intuitions, about linguistic cases for instance. To be able to formally manipulate nouns and verbs in sentences like "The cat sat on the mat" does not automatically endow us with an understanding of what the same nouns and verbs mean. Similarly, the formal manipulation of Chinese symbols in the Chinese Room may not necessarily lead to an understanding of the meaning of these symbols.

When syntax and semantics lie at the same level, the objects of syntactic manipulation—atomic tokens—and the objects of semantic interpretation coincide. We may therefore regard the above reformulation of Searle's axiom as equivalent to the statement: "Manipulation of atomic tokens is never sufficient to endow these tokens with meaning." This statement seems to be at the heart of the intuitions behind Searle's argument.

With this formulation of the relationship between syntax and semantics in place, the application to symbolic and subsymbolic AI is quite clear. By definition, in symbolic models the computational (syntactic) and the representational (semantic) levels coincide. When a symbolic system such as LISP manipulates an expression such as **(IS-A CLYDE ELEPHANT)**, both the syntax and the intended semantics lie at the level of the LISP atom. From our reformulated axiom, if correct, it then follows that the semantics at this level cannot be true semantics. At best, it is extrinsic semantics, thrust upon the symbols by observers. In a connectionist system, on the other hand, the computational and representational levels are quite separate. The fact that there is syntactic manipulation going on at the level of the individual node does not stop there being semantic content at the level of the distributed representation any more than the fact that the cells in a human brain obey iron-clad laws of physics stops there being semantic content at the level of the concept. Because the levels of syntax and semantics are distinct, connectionist networks are safe from the argument.

4.4 Symbolic Replies

The argument above, despite its refinement, is not necessarily a knockdown argument against symbolic models. These have various resources up their sleeve. I examine a few objections that proponents of symbolic models might make, some perhaps valid, some invalid.

1. Symbols in a symbolic system do not get their semantics by being manipulated, but rather by being connected to the world in the right way.

Response: As we have seen, this can give only an account of externalist content, and not an account of internal content (such as conscious mental content), and internal content is what is at stake here. If a system possesses internal content, it does so solely in virtue of its intrinsic properties, and the presence or absence of particular objects in the environment makes no difference. Even a brain in a vat might possess internal content. So this reply is not relevant.

2. Even in a symbolic system, it is not necessarily true that syntax lies at the same level as the semantics. A LISP atom, for instance, may consist of a complex pattern of bits in the system's machine language. The bit-level syntax lies below the atom-level semantics.

Response: As we saw earlier, the computation at the level of the bits is not *causally efficacious* computation. The bits are manipulated as a chunk, and merely serve as an arbitrary label for the atom. They are no more intrinsically important to the atom's functional role within the system than are the individual letters "C", "H", "A", "I", and "R" in the word "CHAIR" relevant to the word's semantic function within language. Both the word and the atom effectively enter into computation as atomic wholes. In contrast, the pattern of activity within a distributed representation is causally efficacious: each node plays an autonomous role. There is no way in which the representation can be interpreted as atomic. In the symbolic case, the bit-level story is mere implementational detail; in the connectionist case, the node-level story plays a vital role.

3. Perhaps the above argument applies to atomic representations in symbolic systems, but what of the complex expressions that are formed from such atoms? Like distributed representations, these have complex internal structure, and thus might possess intrinsic content.

Response: There is something to this argument. It is true that the compositional properties of an expression such as **(IS-A CLYDE ELEPHANT)** are represented by intrinsic features of the representation. The trouble is that *only* the compositional properties are so represented. Such a complex expression, if it has meaning at all, derives most of its meaning from its components, and as we have seen, the components—that is, the atoms **CLYDE** and so on—do not possess intrinsic meaning. Consequently, neither can the complex expression possess full intrinsic meaning. The fact that the expression is about *elephants*, rather than apples, is not represented by any intrinsic feature of the representation. Only the compositional structure of the proposition “Clyde is an elephant” is represented intrinsically, and although this is something, it is not enough.

4. Perhaps symbolic representations do not possess *intrinsic* content, that is, content intrinsic to the tokens themselves. Nevertheless, it is possible that the tokens possess content by virtue of the way in which they interact with each other. Such content would be extrinsic to the individual tokens, but still intrinsic to the system as a whole.

Response: This is the most promising symbolic reply, I believe. Perhaps the requirement that content be represented by internal properties of the representation itself was indeed too stringent. Rather, what is required is that content be represented internally to the system as a whole; but this is consistent with content being extrinsic to individual tokens. Indeed, the theory of functional-role semantics (Block, 1986) suggests that tokens acquire their meaning in virtue of their interactions with the rest of the system. This is certainly a coherent possibility, although it might require a slight rethinking of the symbolic commitments. In particular, if content accrues to a system only in virtue of a particular pattern of causal interactions between tokens, then what sense is there in ascribing content to single tokens alone? It might seem more logical to ascribe content to the patterns of interaction themselves. If we did this, the problem that symbolic representations lack internal structure would be removed. These patterns of interaction would possess internal structure not unlike that of a connectionist pattern of activation.

This reconstrual of symbolic systems might seem to be one way to bridge the conceptual gap between symbolic and connectionist systems. It would, however, require a rethinking of such fundamentals as the Physical Symbol System Hypothesis. If we hold that representation is carried by a pattern of interaction, and not by individual tokens, then the notion that atomic tokens designate would have to be stricken from the hypothesis, in favor of a more complex notion of representation. In some ways, this rethinking might move the foundations of the symbolic endeavor closer to those of the connectionist endeavor, but this might not be such a bad thing.

5 Further Remarks

5.1 Symbol Grounding

The argument I have put forward is quite closely related to the enterprise of *symbol grounding*, as described by Harnad (1990). The project of symbol grounding starts from the observation that computation consists in the manipulation of meaningless symbols. The meaning of these symbols, it is held, is only projected onto them by an observer. For the symbols to possess true semantic content, they must somehow be grounded in some non-symbolic base.

In talking about symbol grounding, one must be careful here not to fall into the old trap of conflating the two quite separate meanings of "symbol", that is, "computational token" and "representation". When one argues that computation consists in the manipulation of meaningless symbols, this is a point about computational tokens. When one asks how symbols can be grounded, this is essentially a question about representations. Computational tokens do not necessarily need to be grounded, as they are not necessarily the object of semantic interpretation in the first place. Harnad (1990) draws the conclusion that no purely computational system can understand, but this may be due to the conflation of computational tokens with representations. The fact that no computational token is grounded does not immediately imply that no representation is grounded.[5] It might therefore be clearer to talk about the *representation grounding* problem: how can a representation in a computational system possess true meaning?

This problem is precisely the problem we have been talking about in this chapter. As the problem stands, there are two possible answers, depending on whether the kind of meaning in question is an internalist or an externalist notion. If one is concerned with how our symbols can possess externalist, extensional content, then one will engage in the project of *causal grounding*. This is the enterprise of hooking up computational systems to the external environment, and thus connecting representations directly to their referents. Proponents of causal grounding hold that there can be no reference in a vacuum, and true representation must be grounded in sensorimotor interaction with the environment. For instance, if the representation **DOG** is triggered by the presence of actual dogs in the environment, then one might fairly claim that the representation really does have *dog* as its external referent.

Causal grounding is an extremely interesting project, but it is not quite so central to this chapter due to its concern with externalist, extensional content. When one asks the question "How can representations be grounded?" about internal content, a different project suggests itself. This is the project of *internal grounding*: ensuring that our representations have sufficient internal structure for them to carry intrinsic content. This is the project that immediately suggests itself from a consideration of symbolic models. In such models, the basic representational units are computational tokens, and such tokens have all the problems with lack

[5] This conflation, incidentally, may also be responsible for much of the appeal of Searle's argument as an argument against the entire AI endeavor. Superficially, there is something quite compelling about the argument: "Computers can only engage in formal symbol manipulation. These symbols are meaningless. Therefore computers cannot understand." But such an argument draws its force entirely from the conflation of representations with computational tokens, under the term "symbol".

of intrinsic meaning that we have already seen.[6]

The goal of internal grounding, then, is to find a vehicle for representational content that is richer than a primitive computational token, and therefore has some chance of possessing intrinsic content. Harnad (1990) suggests one solution: Ground our representations internally in *sensory icons*. This is perhaps a plausible suggestion, but it may be further than we need to go. An alternative possibility, suggested by connectionism, is to ground our representations in distributed patterns of activity. In this way, as we have seen, the internal structure of a representation can systematically reflect the semantic features that it is intended to represent. Connectionism, then, is highly consistent with the goals of this aspect of the symbol grounding project. Connectionist distributed representations are perhaps the best tools currently available through which we can achieve the grounding of representations in their internal structure.

5.2 Representations as patterns

The roots of the vulnerability of symbolic models to the Chinese Room argument lie in the lack of internal structure in basic symbolic representations. Not only has this atomicity of basic representations long been a core assumption of symbolic AI, but it has also had considerable philosophical support. According to Fodor's (1975) influential theory, the mind is nothing but a computational system operating syntactically on structured representations, which have innate, atomic concepts as building blocks. These building blocks (which are of course intended to be semantically interpretable), form the foundation for the "language of thought."

Searle has not been the only one to oppose this view. Much of the impetus for the connectionist movement came from the view that atomic symbols in such a system must necessarily be brittle, rigid, and even empty. Douglas Hofstadter, who certainly does not accept Searle's main argument, nevertheless finds himself when it comes to atomic symbols:

> "Formal tokens such as 'I' or 'hamburger' are in themselves empty. They do not denote. Nor can they be made to denote in the full, rich, intuitive sense of the term by having them obey some rules." (Hofstadter 1985, p. 645)

Such criticisms are reminiscent of the interpretation of Searle that we have made herein. Unlike Searle, however, Hofstadter does not make the leap to an anti-computationalist position. Instead, he considers how these problems might be constructively addressed within a computationalist

[6] The projects of causal grounding and internal grounding are certainly not competing with each other. Rather, each has a different goal. Through the pursuit of both projects, we might be able to come up with adequate accounts of both externalist and internalist content. Taken together, the projects would thus support a "two-factor" theory of meaning (Block, 1986), which many have argued can be the only kind of complete theory of meaning.

framework.

The emptiness of these such symbolic tokens, according to Hofstadter, arises from their passivity. They are “dead, lifeless” tokens, which are only manipulated by some overlying program. There is nothing about the tokens themselves that indicates their reference. Even Newell and Simon (1976) concede this point (happily, it seems):

> A symbol may be used to designate any expression whatsoever. That is, given a symbol, it is not prescribed a priori what expressions it can designate.

According to Newell and Simon, such symbols get their designation only in virtue of how they are manipulated by a program, and how they interact with the outside world.

Hofstadter envisages instead *active symbols*—representations that carry content in their own right. Such representations would be “statistically emergent” from a computational substrate, and would be rich enough to carry their own meaning with them. They would not be formally manipulated by a computer program, but instead emerge from computational manipulations on a lower level.

Connectionism may not as yet have achieved symbols that are fully active in this sense, but it seems to be moving in the right direction. In the sense in which connectionism has symbols, these are not at all atomic, but instead emerge from a multitude of activity at a lower level. These representations have active internal structure in a distributed pattern of activity. Such a patterned internal activity seems to be an important step toward truly active symbols.[7] The idea of *representation as pattern*, at the heart of the connectionist endeavor, may ultimately prove to be the secret of content. Atomic symbols, lacking any internal structure, are content-free. Connectionist distributed representations, however, have so much information carried by the pattern of the representation itself that they are much more plausible candidates to be meaningful.

Connectionist representations have this rich internal pattern precisely because the syntactic structure lies below the symbolic level. If syntax lies at the level of semantics, then symbols are doomed to be atomic, and mental representations are at best a simple compositional combination of such symbols. It is interesting to note that Fodor and Pylyshyn (1988) base their criticisms of connectionism on the idea that connectionist representations do not possess sufficient internal structure. The correct story, it seems, is precisely the opposite: connectionist representations have so much internal structure that they hold much more promise as true representations than do the simple compositional representations of traditional AI.

[7] Kaplan, Weaver, and French (1990) argue that connectionist representations are not yet fully active, due to their not being functionally autonomous. They also offer suggestions about how truly active representations might be achieved in a connectionist framework.

Fodor and Pylyshyn accuse connectionist models of lacking systematicity in representation. Here, too, it is not impossible that the boot may be on the other foot. Symbolic models, it is true, possess a certain kind of *compositional systematicity*—compositional properties of represented entities are reflected by internal properties of a representation. But recent results seem to demonstrate that connectionist models can possess equal compositional systematicity (Blank, Meeden, and Marshall, chapter 6 in this volume; Chalmers 1990; Pollack 1990). What connectionist models possess, and symbolic models lack, is general *semantic systematicity*. In symbolic models, compositional properties are the only semantic properties that are reflected by internal properties of a representation. The semantic properties of basic (noncompositional) representations are not reflected at all. By contrast, in a connectionist model any semantic attribute of an entity is fair game for reflection in the internal structure of a distributed representation. Consequently, such representations can represent any semantic property systematically.

It is precisely because of this semantic systematicity that connectionist models possess many of their most desirable properties, such as automatic generalization. When a number of examples have been presented to a connectionist network, and an appropriate representational framework has been learned by the tuning of connection strengths, a novel example when presented will be systematically slotted into an appropriate representational form. The internal structure of the representation will be relatively similar to the structure of representations of objects to which this example is semantically similar in relevant respects, and quite different from the structure of semantically different objects. Because of this reflection of semantic properties in internal properties, appropriate behavior is generated for the novel example. Without such semantic systematicity, generalization between semantically similar objects is much more difficult, and indeed automatic generalization is rare in symbolic systems. Such similarity-based generalization is just one example of the advantages achieved by the use of representations with rich internal structure.

6 Conclusion

Searle's original arguments treated the field of artificial intelligence as a single endeavor, and the arguments were put forward as applicable to any computational system. But as we have seen, AI is far from unified, and it is vital to draw careful distinctions between different approaches before issuing blanket statements. The "Syntax is not sufficient for semantics" argument, superficially very appealing, may well apply strongly to traditional, symbolic AI. If one's intuitions about AI are derived entirely from the consideration of traditional models, one might take the argument as a condemnation of the entire field. But this would be to ignore the fact that other approaches are possible.

In this chapter, I have tried to analyze the intuitive force of the argument against AI. The argument plays on a fundamental weakness in computational systems: the fact that primitive tokens cannot carry intrinsic content. Instead of treating this intuition as a destructive argument against AI, however, it is perhaps better to view it as a constructive criticism which AI might be able to overcome. It turns out that AI research, chiefly through the connectionist endeavor, may be well on the way to dealing with this problem. I have argued that if we use representational vehicles that are not primitive tokens, but instead possess rich internal pattern, the problem of intrinsic content might be solved.

We should not pretend that there is no difficulty with the idea that semantics might somehow be emergent from syntax, even if the semantics lie at a much higher level. This is indeed a mystery. It is a mystery with which we have been faced with for years, however, in the guise of the human mind. A priori, it would seem implausible to suggest that a mechanical system such as the brain could possess semantics, have understanding, be conscious. Yet it does. The problem of how such a semantic, mental level can emerge from a mechanical substrate is one of the thorniest aspects of the mind–body problem. It should not be surprising that the emergence of semantics from syntax in computational systems should be equally difficult to understand.

Consequently, we might ask Searle to stop proclaiming that "syntax is never sufficient for semantics," and instead join in the investigation of how, possibly, semantics at a high level might emerge from syntax down below. This question will not be resolved overnight, but it seems that connectionism is playing its part in clarifying the issues.

Acknowledgments

I would like to thank Bob French, Douglas Hofstadter, and two referees for their helpful comments.

References

Aizawa, K., Adams, F. & Fuller, G. (this volume). Rules in programming languages and networks.

Armstrong, D. M. (1970). The nature of mind. In C. V. Borst (Ed.), *The mind/brain identity theory* (pp. 67–79). London: Macmillan.

Blank, D. S., Meeden, L. A., & Marshall, J. B. (this volume). Exploring the symbolic/subsymbolic continuum: A case study of RAAM.

Block, N. (1986). Advertisement for a semantics for psychology. In P. French (Ed.), *Midwest Studies in Philosophy*, Vol. 10 (pp. 615–78). Minneapolis: University of Minnesota Press.

Burge, T. (1984). Individualism in psychology. *Philosophical Review*, *95*, 3–45.

Chalmers, D. J. (1990). Syntactic transformations on distributed representations. *Connection Science*, *2*, 53–62.

Chalmers, D. J. (1991). In and out of the Chinese Room. Manuscript in preparation.

Dennett, D. C. (1982). Beyond belief. In A. Woodfield (Ed.), *Thought and object* (pp. 1–95). Oxford: Oxford University Press.

Dyer, M. G. (1990). Distributed symbol formation and processing in connectionist networks. *Journal of Experimental and Theoretical Artificial Intelligence*, *2*, 215–239.

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

Fodor, J. A. (1980). Methodological solipsism considered as a research strategy in cognitive psychology. *Behavioral and Brain Sciences*, *3*, 63–109.

Fodor, J. A., & Pylyshyn, Z. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition, 28*, 3–71.

Franklin, S., & Garzon, M. (1990). Neural computability. In O. Omidvar (Ed.), *Progress in Neural Networks*, Vol. 1., pp. 127-145. Norwood, NJ: Ablex.

Harnad, S. (1990). The symbol grounding problem. *Physica D*, *42*, 335–46.

Hinton, G. E., McClelland, J. L., & Rumelhart, D. E. (1986). Distributed representations. In D. Rumelhart, J. McClelland, & the PDP Researcch Group, *Parallel Distributed Processing* (pp. 77–109). Cambridge, MA: MIT Press.

Hofstadter, D. R. (1985). Waking up from the Boolean dream, or, subcognition as computation. In *Metamagical themas* (pp. 631–665). New York: Basic Books.

Holland, J. H. (1986). Escaping brittleness: The possibilities of general purpose machine learning algorithms applied to parallel rule-based systems. In R. Michalski, J. Carbonell, & T. Mitchell (Eds.), *Machine Learning* (Vol. 2, pp. 593–623). Los Altos, CA: Morgan Kaufmann.

Kaplan, S., Weaver, M. E., & French, R. M. (1990). Active symbols and internal models: Towards a cognitive connectionism. *AI and Society*, *4*, 51–71.

Lakoff, G. (1987). *Women, fire, and dangerous things.* Chicago: University of Chicago Press.

Lewis, D. (1970). Psychophysical and theoretical identifications. *Australasian Journal of Philosophy*, *50*, 249–258.

Mitchell, M., & Hofstadter, D. R. (1990). The emergence of understanding in a computer model of concepts and analogy-making. *Physica D*, *42*, 322–334.

Newell, A., & Simon, H. A. (1976). Computer science as empirical inquiry. *Communications of the Association for Computing Machinery*, *19*, 113–126.

Penrose, R. (1989). *The emperor's new mind.* Oxford: Oxford University Press.

Pollack, J. B. (1990). Recursive distributed representation. *Artificial Intelligence*, *46*, 77–105.

Putnam, H. (1960). Minds and machines. In S. Hook (Ed.), *Dimensions of mind* (pp. 138–164). New York: New York University Press.

Putnam, H. (1975). The meaning of "meaning". In K. Gunderson (Ed.), *Language, mind, and knowledge* (Minnesota Studies in the Philosophy of Science, Vol. 7, pp. 131–193). Minneapolis: University of Minnesota Press.

Rumelhart, D. E., McClelland, J. L., & the PDP Research Group (1986). *Parallel distributed processing*. Cambridge, MA: MIT Press.

Searle, J. R. (1980a). Minds, brains and programs. *Behavioral and Brain Sciences*, *3*, 417–424.

Searle, J. R. (1980b). Intrinsic intentionality. *Behavioral and Brain Sciences*, *3*, 450–457.

Searle, J. R. (1984). *Minds, brains and science.* Cambridge, MA: Harvard University Press.

Searle, J. R. (1987). Minds and brains without programs. In C. Blakemore & S. Greenfield (Eds.), *Mindwaves* (pp. 209–233). Oxford: Blackwell.

Searle, J. R. (1990). Consciousness, explanatory inversion, and cognitive science. *Behavioral and Brain Sciences*, *13*, 585–596.

Smolensky, P. (1988). On the proper treatment of connectionism. *Behavioral and Brain Sciences*, *11*, 1–74.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, *59*, 433–460.