Chapter 16

Escaping from the Chinese room

Margaret A. Boden

JOHN Searle, in his paper on 'Minds, Brains, and Programs' (1980), argues that computational theories in psychology are essentially worthless. He makes two main claims: that computational theories, being purely formal in nature, cannot possibly help us to understand mental processes; and that computer hardware—unlike neuroprotein—obviously lacks the right causal powers to generate mental processes. I shall argue that both these claims are mistaken.

His first claim takes for granted the widely-held (formalist) assumption that the 'computations' studied in computer science are purely syntactic, that they can be defined (in terms equally suited to symbolic logic) as *the formal manipulation of abstract symbols, by the application of formal rules.* It follows, he says, that formalist accounts—appropriate in explaining the meaningless 'information'-processing or 'symbol'-manipulations in computers—are unable to explain how human minds employ *information* or *symbols* properly so-called. Meaning, or intentionality, cannot be explained in computational terms.

Searle's point here is not that no machine can think. Humans can think, and humans—he allows—are machines; he even adopts the materialist credo that only machines can think. Nor is he saying that humans and programs are utterly incommensurable. He grants that, at some highly abstract level of description, people (like everything else) are instantiations of digital computers. His point, rather, is that nothing can think, mean, or understand *solely* in virtue of its instantiating a computer program.

To persuade us of this, Searle employs an ingenious thought-experiment. He imagines himself locked in a room, in which there are various slips of paper with doodles on them; a window through which people can pass further doodle-papers to him, and through which he can pass papers out; and a book of rules (in English) telling him how to pair the doodles, which are always identified by their shape or form. Searle spends his time, while inside the room, manipulating the doodles according to the rules.

One rule, for example, instructs him that when *squiggle-squiggle* is passed in to him, he should give out *squoggle-squoggle.* The rule-book also provides for more complex sequences of doodle-pairing, where only the first and last steps mention

Margaret A. Boden, 'Escaping from the Chinese Room'. In Margaret A. Boden, ed., *The Philosophy of Artificial Intelligence* (Oxford: Oxford University Press, 1990); first published in *Computer Models of Mind* (Cambridge: Cambridge University Press, 1988).

the transfer of paper into or out of the room. Before finding any rule directly instructing him to give out a slip of paper, he may have to locate a *blongle* doodle and compare it with a *blungle* doodle—in which case, it is the result of this comparison which determines the nature of the doodle he passes out. Sometimes many such doodle-doodle comparisons and consequent doodle-selections have to be made by him inside the room before he finds a rule allowing him to pass anything out.

So far as Searle-in-the-room is concerned, the *squiggles* and *squoggles* are mere meaningless doodles. Unknown to him, however, they are Chinese characters. The people outside the room, being Chinese, interpret them as such. Moreover, the patterns passed in and out at the window are understood by them as *questions* and *answers* respectively: the rules happen to be such that most of the questions are paired, either directly or indirectly, with what they recognize as a sensible answer. But Searle himself (inside the room) knows nothing of this.

The point, says Searle, is that Searle-in-the-room is clearly instantiating a computer program. That is, he is performing purely formal manipulations of uninterpreted patterns: he is all syntax and no semantics.

The doodle-pairing rules are equivalent to the IF-THEN rules, or 'productions', commonly used (for example) in expert systems. Some of the internal doodle-comparisons could be equivalent to what Al workers in natural-language processing call a script—for instance, the restaurant script described by R. C. Schank and R. P. Abelson (1977). In that case, Searle-in-the-room's paper-passing performance would be essentially comparable to the performance of a 'question-answering' Schankian text-analysis program. But 'question-answering' is not question-answering. Searle-in-the-room is not really *answering*: how could he, since he cannot understand the questions? Practice does not help (except perhaps in making the doodle-pairing swifter): if Searle-in-the-room ever escapes, he will be just as ignorant of Chinese as he was when he was first locked in.

Certainly, the Chinese people outside might find it useful to keep Searle-in-the-room fed and watered, much as in real life we are willing to spend large sums of money on computerized 'advice' systems. But the fact that people who already possess understanding may use an intrinsically meaningless formalist computational system to provide what they interpret (*sic*) as questions, answers, designations, interpretations, or symbols is irrelevant. They can do this only if they can externally specify a mapping between the formalism and matters of interest to them. In principle, one and the same formalism might be mappable onto several different domains, so could be used (by people) in answering questions about any of those domains. In itself, however, it would be meaningless—as are the Chinese symbols from the point of view of Searle-in-the-room.

It follows, Searle argues, that no system can understand anything solely in virtue of its instantiating a computer program. For if it could, then Searle-in-the-room would understand Chinese. Hence, theoretical psychology cannot properly be grounded in computational concepts.

Searle's second claim concerns what a proper explanation of understanding would be like. According to him, it would acknowledge that meaningful symbols must be embodied in something having 'the right causal powers' for generating understanding, or intentionality. Obviously, he says, brains do have such causal powers whereas computers do not. More precisely (since the brain's organization could be paralleled in a computer), neuroprotein does whereas metal and silicon do not: the biochemical properties of the brain matter are crucial.

A. Newell's (1980) widely cited definition of 'physical-symbol systems' is rejected by Searle, because it demands merely that symbols be embodied in some material that can implement formalist computations—which computers, admittedly, can do. In Searle's view, no electronic computer can really manipulate symbols, nor really designate or interpret anything at all—*irrespective* of any causal dependencies linking its internal physical patterns to its behaviour. (This strongly realist view of intentionality contrasts with the instrumentalism of D. C. Dennett (1971). For Dennett, an intentional system is one whose behaviour we can explain, predict, and control only by ascribing beliefs, goals, and rationality to it. On this criterion, some *existing* computer programs are intentional systems, and the hypothetical humanoids beloved of science-fiction would be intentional systems *a fortiori*.)

Intentionality, Searle declares, is a biological phenomenon. As such, it is just as dependent on the underlying biochemistry as are photosynthesis and lactation. He grants that neuroprotein may not be the only substances in the universe capable of supporting mental life, much as substances other than chlorophyll may be able (on Mars, perhaps) to catalyse the synthesis of carbohydrates. But he rejects metal or silicon as potential alternatives, even on Mars. He asks whether a computer made out of old beer-cans could possibly *understand*—a rhetorical question to which the expected answer is a resounding 'No!' In short, Searle takes it to be intuitively obvious that the inorganic substances with which (today's) computers are manufactured are essentially incapable of supporting mental functions.

In assessing Searle's two-pronged critique of computational psychology, let us first consider his view that intentionality must be biologically grounded. One might be tempted to call this a positive claim, in contrast with his (negative) claim that purely formalist theories cannot explain mentality. However, this would be to grant it more than it deserves, for its explanatory power is illusory. The biological analogies mentioned by Searle are misleading, and the intuitions to which he appeals are unreliable.

The brain's production of intentionality, we are told, is comparable to photosynthesis—but is it, really? We can define the *products* of photosynthesis, clearly distinguishing various sugars and starches within the general class of carbohydrates, and showing how these differ from other biochemical products such as proteins. Moreover, we not only *know that* chlorophyll supports photosynthesis, we also *understand how* it does so (and *why* various other chemicals cannot). We know that it is a catalyst rather than a raw material; and we can specify the point at which,

and the subatomic process by which, its catalytic function is exercised. With respect to brains and understanding, the case is very different.

Our theory of what intentionality is (never mind how it is generated) does not bear comparison with our knowledge of carbohydrates: just what intentionality *is* is still philosophically controversial. We cannot even be entirely confident that we can recognize it when we see it. It is generally agreed that the propositional attitudes are intentional, and that feelings and sensations are not; but there is no clear consensus about the intentionality of emotions.

Various attempts have been made to characterize intentionality and to distinguish its subspecies as distinct intentional states (beliefs, desires, hopes, intentions, and the like). Searle himself has made a number of relevant contributions, from his early work on speech-acts (1969) to his more recent account (1983) of intentionality in general. A commonly used criterion (adopted by Brentano in the nineteenth century and also by Searle) is a *psychological* one. In Brentano's words, intentional states direct the mind on an object; in Searle's, they have intrinsic representational capacity, or 'aboutness'; in either case they relate the mind to the world, and to possible worlds. But some writers define intentionality in *logical* terms (Chisholm 1967). It is not even clear whether the logical and psychological definitions are precisely co-extensive (Boden 1970). In brief, no theory of intentionality is accepted as unproblematic, as the chemistry of carbohydrates is.

As for the brain's biochemical 'synthesis' of intentionality, this is even more mysterious. We have very good reason to believe *that* neuroprotein supports intentionality, but we have hardly any idea *how—qua* neuroprotein—it is able to do so.

In so far as we understand these matters at all, we focus on the neurochemical basis of certain *informational functions*—such as message-passing, facilitation, and inhibition—embodied in neurones and synapses. For example: how the sodium-pump at the cell-membrane enables an action potential to propagate along the axon; how electrochemical changes cause a neurone to enter into and recover from its refractory period; or how neuronal thresholds can be altered by neurotransmitters, such as acetylcholine.

With respect to a visual cell, for instance, a crucial psychological question may be *whether it can function so as to detect intensity-gradients.* If the neurophysiologist can tell us which molecules enable it to do so, so much the better. But from the psychological point of view, it is not the biochemistry as such which matters but the information-bearing functions grounded in it. (Searle apparently admits this when he says, 'The type of realizations that intentional states have in the brain may be describable at a much higher functional level than that of the specific biochemistry of the neurons involved' (1983: 272).)

As work in 'computer vision' has shown, metal and silicon are undoubtedly able to support some of the functions necessary for the 2D-to-3D mapping involved in vision. Moreover, they can embody specific mathematical functions for recognizing intensity-gradients (namely 'DOG-detectors', which compute the differénce of Gaussians) which seem to be involved in many biological visual systems. Admit-

tedly, it may be that metal and silicon cannot support all the functions involved in normal vision, or in understanding generally. Perhaps only neuroprotein can do so, so that only creatures with a 'terrestrial' biology can enjoy intentionality. But we have no specific reason, at present, to think so. Most important in this context, any such reasons we might have in the future must be grounded in empirical discovery: intuitions will not help.

If one asks which mind-matter dependencies are intuitively plausible, the answer must be that *none* is. Nobody who was puzzled about intentionality (as opposed to action-potentials) ever exclaimed 'Sodium—of course!' Sodium-pumps are no less 'obviously' absurd than silicon chips, electrical polarities no less 'obviously' irrelevant than old beer-cans, acetylcholine hardly less surprising than beer. The fact that the first member of each of these three pairs is *scientifically* compelling does not make any of them *intuitively* intelligible: our initial surprise persists.

Our intuitions might change with the advance of science. Possibly we shall eventually see neuroprotein (and perhaps silicon too) as obviously capable of embodying mind, much as we now see biochemical substances in general (including chlorophyll) as obviously capable of producing other such substances—an intuition that was not obvious, even to chemists, prior to the synthesis of urea. At present, however, our intuitions have nothing useful to say about the material basis of intentionality. Searle's 'positive' claim, his putative alternative explanation of intentionality, is at best a promissory note, at worst mere mystery-mongering.

Searle's negative claim—that formal-computational theories cannot explain understanding—is less quickly rebutted. My rebuttal will involve two parts: the first directly addressing his example of the Chinese room, the second dealing with his background assumption (on which his example depends) that computer programs are pure syntax.

The Chinese-room example has engendered much debate, both within and outside the community of cognitive science. Some criticisms were anticipated by Searle himself in his original paper, others appeared as the accompanying peer-commentary (together with his Reply), and more have been published since. Here, I shall concentrate on only two points: what Searle calls the Robot reply, and what I shall call the English reply.

The Robot reply accepts that the only understanding of Chinese which exists in Searle's example is that enjoyed by the Chinese people outside the room. Searle-in-the-room's inability to connect Chinese characters with events in the outside world shows that he does not understand Chinese. Likewise, a Schankian teletyping computer that cannot recognize a restaurant, hand money to a waiter, or chew a morsel of food understands nothing of restaurants—even if it can usefully 'answer' our questions about them. But a robot, provided not only with a restaurantscript but also with camera-fed visual programs and limbs capable of walking and picking things up, would be another matter. If the input-output behaviour of such a robot were identical with that of human beings, then it would demonstrably understand

both restaurants and the natural language—Chinese, perhaps—used by people to communicate with it.

Searle's first response to the Robot reply is to claim a victory already, since the reply concedes that cognition is not solely a matter of formal symbol-manipulation but requires in addition a set of causal relations with the outside world. Second, Searle insists that to add perceptuomotor capacities to a computational system is not to add intentionality, or understanding.

He argues this point by imagining a robot which, instead of being provided with a computer program to make it work, has a miniaturized Searle inside it—in its skull, perhaps. Searle-in-the-robot, with the aid of a (new) rule-book, shuffles paper and passes *squiggles* and *squoggles* in and out, much as Searle-in-the-room did before him. But now some or all of the incoming Chinese characters are not handed in by Chinese people, but are triggered by causal processes in the cameras and audio-equipment in the robot's eyes and ears. And the outgoing Chinese characters are not received by Chinese hands, but by motors and levers attached to the robot's limbs—which are caused to move as a result. In short, this robot is apparently able not only to answer questions in Chinese, but also to see and do things accordingly: it can recognize raw beansprouts and, if the recipe requires it, toss them into a wok as well as the rest of us.

(The work on computer vision mentioned above suggests that the vocabulary of Chinese would require considerable extension for this example to be carried through. And the large body of AI research on language-processing suggests that the same could be said of the English required to express the rules in Searle's initial 'question-answering' example. In either case, what Searle-in-the-room needs is not so much Chinese, or even English, as a programming-language. We shall return to this point presently.)

Like his roombound predecessor, however, Searle-in-the-robot knows nothing of the wider context. He is just as ignorant of Chinese as he ever was, and has no more purchase on the outside world than he did in the original example. To him, beansprouts and woks are invisible and intangible: all Searle-in-the-robot can see and touch, besides the rule-book and the doodles, are his own body and the inside walls of the robot's skull. Consequently, Searle argues, the robot cannot be credited with understanding of any of these worldly matters. In truth, it is not *seeing* or *doing* anything at all: it is 'simply moving about as a result of its electrical wiring and its program', which latter is instantiated by the man inside it, who 'has no intentional states of the relevant type' (1980: 420).

Searle's argument here is unacceptable as a rebuttal of the Robot reply, because it draws a false analogy between the imagined example and what is claimed by computational psychology.

Searle-in-the-robot is supposed by Searle to be performing the functions performed (according to computational theories) by the human brain. But, whereas most computationalists do not ascribe intentionality to the brain (and those who do, as we shall see presently, do so only in a very limited way), Searle characterizes

Searle-in-the-robot as enjoying full-blooded intentionality, just as he does himself. Computational psychology does not credit the brain with *seeing beansprouts* or *understanding English*: intentional states such as these are properties of people, not of brains. In general, although representations and mental processes are assumed (by computationalists and Searle alike) to be embodied in the brain, the sensorimotor capacities and propositional attitudes which they make possible are ascribed to the person as a whole. So Searle's description of the system inside the robot's skull as one which can understand English does not truly parallel what computationalists say about the brain.

Indeed, the specific procedures hypothesized by computational psychologists, and embodied by them in computer models of the mind, are relatively stupid—and they become more and more stupid as one moves to increasingly basic theoretical levels. Consider theories of natural-language parsing, for example. A parsing procedure that searches for a determiner does not understand English, and nor does a procedure for locating the reference of a personal pronoun: only the person whose brain performs these interpretive processes, and many others associated with them, can do that. The capacity to understand English involves a host of interacting information processes, each of which performs only a very limited function but which together provide the capacity to take English sentences as input and give appropriate English sentences as output. Similar remarks apply to the individual components of computational theories of vision, problem-solving, or learning. Precisely because psychologists wish to *explain* human language, vision, reasoning, and learning, they posit underlying processes which lack the capacities.

In short, Searle's description of the robot's pseudo-brain (that is, of Searle-in-the-robot) as understanding English involves a category-mistake comparable to treating the brain as the bearer—as opposed to the causal basis—of intelligence.

Someone might object here that I have contradicted myself, that I am claiming that one cannot ascribe intentionality to brains and yet am implicitly doing just that. For I spoke of the brain's effecting 'stupid' component-procedures—but stupidity is virtually a *species* of intelligence. To be stupid is to be intelligent, but not very (a person or a fish can be stupid, but a stone or a river cannot).

My defence would be twofold. First, the most basic theoretical level of all would be at the neuroscientific equivalent of the machine-code, a level 'engineered' by evolution. The facts that a certain light-sensitive cell *can* respond to intensity-gradients by acting as a DOG-detector and that one neurone *can* inhibit the firing of another, are explicable by the biochemistry of the brain. The notion of stupidity, even in scare-quotes, is wholly inappropriate in discussing such facts. However, these very basic information-processing functions (DOG-detecting and synaptic inhibition) *could* properly be described as 'very, very, very ... stupid'. This of course implies that intentional language, if only of a highly grudging and uncomplimentary type, is applicable to brain processes after all—which prompts the second point in my defence. I did not say that intentionality cannot be ascribed to brains, but that full-blooded intentionality cannot. Nor did I say that brains

cannot understand anything at all, in howsoever limited a fashion, but that they cannot (for example) understand English. I even hinted, several paragraphs ago, that a few computationalists do ascribe some degree of intentionality to the brain (or to the computational processes going on in the brain). These two points will be less obscure after we have considered the English reply and its bearing on Searle's background assumption that formal-syntactic computational theories are purely syntactic.

The crux of the English reply is that the instantiation of a computer program, whether by man or by manufactured machine, does involve understanding—at least of the rule-book. Searle's initial example depends critically on Searle-in-the-room's being able to understand the language in which the rules are written, namely English; similarly, without Searle-in-the-robot's familiarity with English, the robot's beansprouts would never get thrown into the wok. Moreover, as remarked above, the vocabulary of English (and, for Searle-in-the-robot, of Chinese too) would have to be significantly modified to make the example work.

An unknown language (whether Chinese or Linear B) can be dealt with only as an aesthetic object or a set of systematically related forms. Artificial languages can be designed and studied, by the logician or the pure mathematician, with only their structural properties in mind (although D. R. Hofstadter's (1979) example of the quasi-arithmetical pq-system shows that a psychologically compelling, and predictable, interpretation of a formal calculus may arise spontaneously). But one normally responds in a very different way to the symbols of one's native tongue; indeed, it is very difficult to 'bracket' (ignore) the meanings of familiar words. The view held by computational psychologists, that natural languages can be characterized in procedural terms, is relevant here: words, clauses, and sentences can be seen as mini-programs. The symbols in a natural language one understands initiate mental activity of various kinds. To learn a language is to set up the relevant causal connections, not only between words and the world ('cat' and the thing on the mat) but between words and the many non-introspectible procedures involved in interpreting them.

Moreover, we do not need to be told *ex hypothesi* (by Searle) that Searle-in-the-room understands English: his behaviour while in the room shows clearly that he does. Or, rather, it shows that he understands a *highly limited subset* of English.

Searle-in-the-room could be suffering from total amnesia with respect to 99 per cent of Searle's English vocabulary, and it would make no difference. The only grasp of English he needs is whatever is necessary to interpret (*sic*) the rule-book—which specifies how to accept, select, compare, and give out different patterns. Unlike Searle, Searle-in-the-room does not require words like 'catalyse', 'beer-can', chlorophyll', and 'restaurant'. But he may need 'find', 'compare', 'two', 'triangular', and 'window' (although his understanding of these words could be much less full than Searle's). He must understand conditional sentences, if any rule states that if he sees a *squoggle* he should give out a *squiggle*. Very likely, he must understand some way of expressing negation, temporal ordering, and (especially if he is to learn to do his job faster) generalization. If the rules he uses include some which parse

the Chinese sentences, then he will need words for grammatical categories too. (He will not need explicit rules for parsing English sentences, such as the parsing procedures employed in AI programs for language-processing, because he already understands English.)

In short, Searle-in-the-room needs to understand only that subset of Searle's English which is equivalent to the programming-language understood by a computer generating the same 'question-answering' input-output behaviour at the window. Similarly, Searle-in-the-robot must be able to understand whatever subset of English is equivalent to the programming-language understood by a fully computerized visuomotor robot.

The two preceding sentences may seem to beg the very question at issue. Indeed, to speak thus of the programming-language understood by a computer is seemingly self-contradictory. For Searle's basic premiss—which he assumes is accepted by all participants in the debate—is that a computer program is purely formal in nature: the computation it specifies is purely syntactic and has no intrinsic meaning or semantic content to be understood.

If we accept this premiss, the English reply sketched above can be dismissed forthwith for seeking to draw a parallel where no parallel can properly be drawn. But if we do not, if—*pace* Searle (and others (Fodor 1980; Stich 1983))—computer programs are not concerned only with syntax, then the English reply may be relevant after all. We must now turn to address this basic question.

Certainly, one can for certain purposes think of a computer program as an uninterpreted logical calculus. For example, one might be able to prove, by purely formal means, that a particular well-formed formula is derivable from the program's data-structures and inferential rules. Moreover, it is true that a so-called interpreter program that could take as input the list-structure '(FATHER (MAGGIE))' and return '(LEONARD)' would do so on formal criteria alone, having no way of interpreting these patterns as possibly denoting real people. Likewise, as Searle points out, programs provided with restaurant-scripts are not thereby provided with knowledge of restaurants. The existence of a mapping between a formalism and a certain domain does not in itself provide the manipulator of the formalism with any understanding of that domain.

But what must not be forgotten is that a computer program is *a program for a computer*: when a program is run on suitable hardware, the machine *does* something as a result (hence the use in computer science of the words 'instruction' and 'obey'). At the level of the machine-code the effect of the program on the computer is direct, because the machine is engineered so that a given instruction elicits a unique operation (instructions in high-level languages must be converted into machine-code instructions before they can be obeyed). A programmed instruction, then, is not a mere formal pattern—nor even a declarative statement (although it may for some purposes be thought of under either of those descriptions). It is a procedure specification that, given a suitable hardware context, can cause the procedure in question to be executed.

One might put this by saying that a programming-language is a medium not only for expressing *representations* (structures that can be written on a page or provided to a computer, some of which structures may be isomorphic with things that interest people) but also for bringing about the *representational activity* of certain machines.

One might even say that a representation *is* an activity rather than a structure. Many philosophers and psychologists have supposed that mental representations are intrinsically active. Among those who have recently argued for this view is Hofstadter (1985: 648), who specifically criticizes Newell's account of *symbols* as manipulable formal tokens. In his words, 'The brain itself does not 'manipulate symbols'; the brain is the medium in which the symbols are floating and in which they trigger each other.' Hofstadter expresses more sympathy for 'connectionist' than for 'formalist' psychological theories. Connectionist approaches involve parallel-processing systems broadly reminiscent of the brain, and are well suited to model cerebral representations, symbols, or concepts, as *dynamic*. But it is not only connectionists who can view concepts as intrinsically active, and not only *cerebral* representations which can be thought of in this way: this claim has been generalized to cover traditional computer programs, specifically designed for von Neumann machines. The computer scientist B. C. Smith (1982) argues that programmed representations, too, are inherently active—and that an adequate theory of the semantics of programming-languages would recognize the fact.

At present, Smith claims, computer scientists have a radically inadequate understanding of such matters. He reminds us that, as remarked above, there is no general agreement—either within or outside computer science—about what *intentionality* is, and deep unclarities about *representation* as well. Nor can unclarities be avoided by speaking more technically, in terms of *computation* and *formal symbol-manipulation*. For the computer scientist's understanding of what these phenomena really are is also largely intuitive. Smith's discussion of programming-languages identifies some fundamental confusions within computer science. Especially relevant here is his claim that computer scientists commonly make too complete a theoretical separation between a program's control-functions and its nature as a formal-syntactic system.

The theoretical divide criticized by Smith is evident in the widespread 'dual-calculus' approach to programming. The dual-calculus approach posits a sharp theoretical distinction between a declarative (or denotational) representational structure and the procedural language that interprets it when the program is run. Indeed, the knowledge-representation and the interpreter are sometimes written in two quite distinct formalisms (such as predicate calculus and LISP, respectively). Often, however, they are both expressed in the same formalism; for example, LISP (an acronym for LISt-Processing language) allows facts and procedures to be expressed in formally similar ways, and so does PROLOG (PROgramming-in-LOGic). In such cases, the dual-calculus approach dictates that the (single) programming-language concerned be theoretically described in two quite different ways.

To illustrate the distinction at issue here, suppose that we wanted a representation of family relationships which could be used to provide answers to questions about such matters. We might decide to employ a list-structure to represent such facts as that Leonard is the father of Maggie. Or we might prefer a frame-based representation, in which the relevant name-slots in the FATHER-frame could be simultaneously filled by 'LEONARD' and 'MAGGIE'. Again, we might choose a formula of the predicate calculus, saying that there exist two people (namely, Leonard and Maggie), and Leonard is the father of Maggie. Last, we might employ the English sentence 'Leonard is the father of Maggie.'

Each of these four representations could be written/drawn on paper (as are the rules in the rule-book used by Searle-in-the-room), for us to interpret *if* we have learnt how to handle the relevant notation. Alternatively, they could be embodied in a computer database. But to make them usable by the computer, there has to be an interpreter-program which (for instance) can find the item 'LEONARD' when we 'ask' it who is the father of Maggie. No one with any sense would embody list-structures in a computer without providing it also with a *list-processing* facility, nor give it frames without a *slot-filling* mechanism, logical formulae without *rules of inference*, or English sentences without *parsing procedures*. (Analogously, people who knew that Searle speaks no Portuguese would not give Searle- in-the-room a Portuguese rule-book unless they were prepared to teach him the language first.)

Smith does not deny that there is an important distinction between the *denotational import* of an expression (broadly: what actual or possible worlds can be mapped onto it) and its *procedural consequence* (broadly: what it does, or makes happen). The fact that the expression '(FATHER (MAGGIE))' is isomorphic with a certain parental relationship between two actual people (and so might be mapped onto that relationship by us) is one thing. The fact that the expression '(FATHER (MAGGIE))' can cause a certain computer to locate 'LEONARD' is quite another thing. Were it not so, the dual-calculus approach would not have developed. But he argues that, rather than persisting with the dual-calculus approach, it would be more elegant and less confusing to adopt a 'unified' theory of programming-languages, designed to cover both denotative and procedural aspects.

He shows that many basic terms on either side of the dual-calculus divide have deep theoretical commonalities as well as significant differences. The notion of *variable*, for instance, is understood in somewhat similar fashion by the logician and the computer scientist: both allow that a variable can have different *values* assigned to it at different times. That being so, it is redundant to have two distinct theories of what a variable is. To some extent, however, logicians and computer scientists understand different things by this term: the value of a variable in the LISP programming-language (for example) is another LISP-expression, whereas the value of a variable in logic is usually some object external to the formalism itself. These differences should be clarified—not least to avoid confusion when a system attempts to reason *about* variables by *using* variables. In short, we need a single definition of 'variable', allowing both for its declarative use (in logic) and for

its procedural use (in programming). Having shown that similar remarks apply to other basic computational terms, Smith outlines a unitary account of the semantics of LISP and describes a new calculus (MANTIQ) designed with the unified approach in mind.

As the example of using variables to reason about variables suggests, a unified theory of computation could illuminate how *reflective* knowledge is possible. For, given such a theory, a system's representations of data and of processes—including processes internal to the system itself—would be essentially comparable. This theoretical advantage has psychological relevance (and was a major motivation behind Smith's work).

For our present purposes, however, the crucial point is that a fundamental theory of *programs*, and of *computation*, should acknowledge that an essential function of a computer program is to make things happen. Whereas symbolic logic can be viewed as mere playing around with uninterpreted formal calculi (such as the predicate calculus), and computational logic can be seen as the study of abstract timeless relations in mathematically specified 'machines' (such as Turing machines), computer science cannot properly be described in either of these ways.

It follows from Smith's argument that the familiar characterization of computer programs as all syntax and no semantics is mistaken. The inherent procedural consequences of any computer program give it a toehold in semantics, where the semantics in question is not denotational, but causal. The analogy is with Searle-in-the-room's understanding of English, not his understanding of Chinese.

This is implied also by A. Sloman's (1986*a*; 1986*b*) discussion of the sense in which programmed instructions and computer symbols must be thought of as having some semantics, however restricted. In a causal semantics, the meaning of a symbol (whether simple or complex) is to be sought by reference to its causal links with other phenomena. The central questions are 'What causes the symbol to be built and/or activated?' and 'What happens as a result of it?' The answers will sometimes mention external objects and events visible to an observer, and sometimes they will not.

If the system is a human, animal, or robot, it may have causal powers which enable it to refer to restaurants and beansprouts (the philosophical complexities of reference to external, including unobservable, objects may be ignored here, but are helpfully discussed by Sloman). But whatever the information-processing system concerned, the answers will sometimes describe purely *internal* computational processes—whereby other symbols are built, other instructions activated. Examples include the interpretative processes inside Searle-in-the-room's mind (comparable perhaps to the parsing and semantic procedures defined for automatic natural-language processing) that are elicited by English words, and the computational processes within a Schankian text-analysis program. Although such a program cannot use the symbol 'restaurant' to mean *restaurant* (because it has no causal links with restaurants, food and so forth), its internal symbols and procedures do

embody some minimal understanding of certain other matters—of what it is to compare two formal structures, for example.

One may feel that the 'understanding' involved in such a case is *so* minimal that this word should not be used at all. So be it. As Sloman makes clear, the important question is not '*When does a machine understand something?*' (a question which misleadingly implies that there is some clear cut-off point at which understanding ceases) but '*What things does a machine (whether biological or not) need to be able to do in order to be able to understand?*' This question is relevant not only to the *possibility* of a computational psychology, but to its *content* also.

In sum, my discussion has shown Searle's attack on computational psychology to be ill founded. To view Searle-in-the-room as an instantiation of a computer program is not to say that he lacks all understanding. Since the theories of a formalist-computational psychology should be likened to computer programs rather than to formal logic, computational psychology is not in principle incapable of explaining how meaning attaches to mental processes.

References

Boden, M. A (1970). 'Intentionality and Physical Systems.' *Philosophy of Science* 37: 200–14.

Chisholm, R. M. (1967). 'Intentionality.' In P. Edwards (ed.), *The Encyclopedia of Philosophy.* Vol. IV, pp. 201–4. New York: Macmillan.

Dennett, D. C. (1971). 'Intentional Systems.' *J. Philosophy* 68: 87–106. Repr. in D. C. Dennett, *Brainstorms: Philosophical Essays on Mind and Psychology*, pp. 3–22. Cambridge, Mass.: MIT Press, 1978.

Fodor, J. A. (1980). 'Methodological Solipsism Considered as a Research Strategy in Cognitive Psychology.' *Behavioral and Brain Sciences* 3: 63–110. Repr. in J. A. Fodor, *Representations: Philosophical Essays on the Foundations of Cognitive Science*, pp. 225–56. Brighton: Harvester Press, 1981.

Hofstadter, D. R. (1979). *Godel, Escher, Bach: An Eternal Golden Braid.* New York: Basic Books.

——(1985). 'Waking Up from the Boolean Dream; Or, Subcognition as Computation.' In D. R. Hofstadter, *Metamagical Themas: Questing for the Essence of Mind and Pattern*, pp. 631–65. New York: Viking.

Newell, A. (1980). 'Physical Symbol Systems.' *Cognitive Science* 4: 135–83.

Schank, R. C., and Abelson, R. P. (1977). *Scripts, Plans, Goals, and Understanding.* Hillsdale, NJ: Erlbaum.

Searle, J. R. (1969). *Speech Acts: An Essay in the Philosophy of Language.* Cambridge: Cambridge University Press.

——(1980). 'Minds, Brains, and Programs.' *Behavioral and Brain Sciences* 3: 417–24. (See Chapter 15 of this volume.)

——(1983). *Intentionality: An Essay in the Philosophy of Mind.* Cambridge: Cambridge University Press.

Sloman, A. (1986*a*). 'Reference Without Causal Links.' In B. du Boulay and L. J. Steels (eds.), *Seventh European Conference on Artificial Intelligence*, pp. 369–81. Amsterdam: North-Holland.

Sloman, A. (1986*b*). 'What Sorts of Machines Can Understand the Symbols They Use?' *Proc. Aristotelian Soc.* Supp. 60: 61–80.

Smith, B. C. (1982). *Reflection and Semantics in a Procedural Language.* Cambridge, Mass.: MIT Ph.D. dissertation and Technical Report LCS/TR-272.

Stich, S. C. (1983). *From Folk Psychology to Cognitive Science: The Case Against Belief.* Cambridge, Mass.: MIT Press/Bradford Books.

Philosophy of Mind

A Guide and Anthology

John Heil

OXFORD
UNIVERSITY PRESS

OXFORD
UNIVERSITY PRESS

Great Clarendon Street, Oxford OX2 6DP

Oxford University Press is a department of the University of Oxford.
It furthers the University's objective of excellence in research, scholarship,
and education by publishing worldwide in

Oxford New York

Auckland Bangkok Buenos Aires Cape Town Chennai
Dar es Salaam Delhi Hong Kong Istanbul Karachi Kolkata
Kuala Lumpur Madrid Melbourne Mexico City Mumbai Nairobi
São Paulo Shanghai Taipei Tokyo Toronto

Oxford is a registered trade mark of Oxford University Press
in the UK and in certain other countries

Published in the United States
by Oxford University Press Inc., New York

First published in 2004

British Library Cataloguing in Publication Data

Data available

Library of Congress Cataloging in Publication Data

Data available

ISBN 0–19–925383–8

10 9 8 7 6 5 4 3 2 1

Typeset in Adobe Minion by RefineCatch Limited, Bungay, Suffolk
Printed in Great Britain by TJ International Ltd, Padstow, Cornwall