BULLETIN (New Series) OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 2, Number 2, March 1980

0002-9904/80/0000-0109/$03.75

The computer revolution in philosophy: *Philosophy, science and models of mind*, by Aaron Sloman, Harvester Studies in Cognitive Science Humanities Press, Atlantic Highlands, N. J., 1978, xvi + 304 pp., cloth, $22.50.

Aaron Sloman is a man who is convinced that most philosophers and many other students of mind are in dire need of being convinced that there has been a revolution in that field happening right under their noses, and that they had better quickly inform themselves. The revolution is called "Artificial Intelligence" (AI)–and Sloman attempts to impart to others the "enlightenment" which he clearly regrets not having experienced earlier himself. Being somewhat of a convert, Sloman is a zealous campaigner for his point of view. Now a Reader in Cognitive Science at Sussex, he began his academic career in more orthodox philosophy and, by exposure to linguistics and AI, came to feel that all approaches to mind which ignore AI are missing the boat. I agree with him, and I am glad that he has written this provocative book.

The tone of Sloman's book can be gotten across by this quotation (p. 5): "I am prepared to go so far as to say that within a few years, if there remain any philosophers who are not familiar with some of the main developments in artificial intelligence, it will be fair to accuse them of professional incompetence, and that to teach courses in philosophy of mind, epistemology, aesthetics, philosophy of science, philosophy of language, ethics, metaphysics, and other main areas of philosophy, without discussing the relevant aspects of artificial intelligence will be as irresponsible as giving a degree course in physics which includes no quantum theory."

Strong language–but coming from a former philosopher, it bears listening to. Unfortunately, the harsh style of the book–its occasionally condescending tone and brusque dismissal of many points of view–although often justified, will likely lose him many of the readers he hopes to convert. I offer a couple of polemical selections below:

> Shallow philosophical, linguistic, and psychological disputes about innate or non-empirical knowledge are being replaced by much harder and deeper explorations of exactly what pre-existing knowledge is required, or sufficient, for particular types of empirical and non-empirical learning. (p. 11)

> I shall ignore the many pseudo-questions posed by incompetent philosophers who cannot tell the difference between profundity and obscurity. (p. 65)

> In future, if philosophers and psychologists wish to avoid superficiality and triviality in their studies of abstract or general concepts, they will need to be informed about the versions of Kant's idea which are being explored in attempts to make computers intelligent. (p. 185)

> In my experience, philosophers and psychologists tend to get very confused about how to deal with this kind of circularity. (p. 211)

> The typical style of philosophical discussion . . . all too often is a mixture of dubious introspective reports, dualist or anti-dualist prejudice, and pompous confused terminology. (p. 252)

The principal targets of his jabs are, clearly, philosophers and psychologists–but neurophysiologists, educators, logicians, biologists, psychiatrists, social scientists, smokers, administrators, historians, students of literature, and popular science journalists do not escape his attack. Linguists receive mixed reviews.

Perhaps in Britain it is the custom to be so intellectually aggressive but personally I find such polemicizing gratuitous and offensive, despite my intellectual agreement with most of his points.

But enough complaining. The author has done an outstanding job of showing how AI, more than any amount of philosophical theorizing, will shed light–indeed, already has–on all the traditional quandaries about knowledge, memory, meaning, perception, consciousness, free will, learning, creativity, and so on. The key idea he is propounding, then, is (p. 3), "to view complex processes as computational processes, including rich information flow between subprocesses and the construction and manipulating of symbolic structures within processes."

Sloman clearly has had many a conversation with doubting colleagues and friends, for he takes pains to dispel false first impressions which many people have about computers, AI, etc. For example, he draws a sharp distinction between AI and (1) ordinary scientific programming ("number-crunching", as it is sometimes called) in "lower-level" languages like Fortran, (2) cybernetics and "systems theory", (3) Shannon and Weaver's information theory, (4) mathematical modeling in psychology, and so on.

Perhaps the most common misconception–or rather, oversimplification–about computers was first formulated in 1841 by Lady Ada Lovelace, and is restated by Sloman this way: "However complex the programs that run in them [computers] they are always essentially unintelligent, uncreative mechanisms, blindly following simple rules one at a time." (p. 112) He provides the following partial answers:

> Such a description may well be true of the underlying electronic components, just as it may well be true to say that a human brain is always essentially an unintelligent uncreative bundle of nerve-cells (or an assemblage of atoms) blindly reacting to one another in accordance with chemical and physical laws of nature. But just as the latter description may omit some important features of what a brain can do, so also the former description omits important "high-level" features of complex computer programs. What is true of a computer need not be true of a program, just as what is true of a brain need not be true of a mind. In both cases the whole is far more than the sum of its parts. (p. 112)

If only he had left that last sentence out, I would have been much happier. AI seems to be riding a current wave of "antireductionistic" sentiment, despite the fact that in truth, AI represents the epitome of the reductionistic view of mind. Although I know what he means by this overworked cliché, I wish he would not lend fuel to the simplistic, holistic school of mind. In any case, more of his answer is provided by this remark (p. 105):

> Complex programs sometimes work for reasons which their designers only half understand, and often they fail in ways which their designers cannot understand. It follows that nobody is in a position to make pronouncements about the limits of what can be done by computer programs, especially

> programs which interact with some complex environment, as people do.
>
> Attempting such pronouncements is about as silly as attempting to use an analysis of the printing process to delimit the kinds of theories that will be expounded in textbooks of physics in a hundred years time. Nevertheless, people with theological or other motives for believing that computers cannot match human beings will continue to be overconfident about such matters (e.g. H. Dreyfus, *What computers can't do*).

We will come back to the issue of free will and the limits of programs later.

A main thesis of the book is that, even if philosophers and psychologists themselves do not become AI workers, they can benefit immeasurably merely from acquiring familiarity with its concepts and others from computer science in general, such as:

> parsing, compiling, interpreting, pointer, mutual recursion, push, pop, stack, tree, threaded list, address, environment, data abstraction, real time, CPU time, machine architecture, paging, computational complexity, return address, instruction set, microcode, sharable code, debugging, deadlock, thrashing, sequential memory, random access memory, fast memory, slow memory, side effect, pattern matching, interrupt, high-level language, machine language, data structure, procedure, control structure, coroutine, backtracking, demon, virtual machine, heterarchy, operating system, bootstrapping, circular list, etc. etc.

Before he presents his refreshingly idiosyncratic view of present and future AI, Sloman devotes several chapters to a careful analysis of scientific explanation, with the eventual aim of demonstrating exactly why doing AI is a valid kind of science. To my mind, these chapters are not nearly as interesting as the later ones on AI itself. Perhaps that is because to me, good AI speaks for itself, without need of verbose philosophical defenses–the very thing which Sloman so often derides. Yet it is quite conceivable to me that Sloman, by articulately speaking the philosophers' language, can reach them.

Incidentally, related to the question of whether AI is "good science" is this one: "Is an AI program a *mathematical* theory of mind?" Although ultimately it probably is only a matter of semantics, it is still worth exploring for a moment. Many mathematicians and computer people tend naïvely to view a program of any degree of complexity as a mathematical object, probably because it is, in some not-terribly-clear sense, a Turing-computable function of its inputs. However true this may be in a theoretical sense, in fact it is simply irrelevant. The idea of proving programs correct is now a subdiscipline of computer science, but it has not advanced very far. As for proving an AI program "correct" (not that it is clear what this would mean), Sloman has this to say (p. 141):

> it is necessary either to analyse the specifications of the mechanisms and of the possibilities to be explained, and then prove mathematically that the mechanism does generate the required range of possibilities and nothing which it should not generate, *or else* to construct the mechanism and run it experimentally in a wide variety of circumstances to ensure that it produces an adequate variety of behaviour, with the required fine structure.
>
> The former is likely to be well beyond the possibilities of mathematical analysis available in the foreseeable future, even though the mathematical analysis of programs and proof of their correctness is a developing discipline. In particular, it assumes that we can produce complete specifications of the possibilities to be explained, whereas one of the lessons of artificial

> intelligence is that attempting to design a working system often leads you to revise and extend your specifications.

A recent article, *Social Processes and Proofs of Theorems and Programs* (De Millo, Lipton, and Perlis, Comm. ACM **22** (1979)) decries the notion of program verification, shows how greatly it differs from that of mathematical proof, and urges programmers to rely on experience and common sense, not formal proofs. A far cry from seeing programs as "mathematical objects"! Is it correct, then, to look upon AI as a "mathematical approach to mind"?

The answer is 'no', in that AI is not based on equations, deductive theories, or pristinely elegantly structures; the answer is 'yes', however, in that AI is *formal*. In fact this leads to a curious sort of seeming paradox: How can a totally formal (hence rigid) system be flexible and creative, as the mind is? This paradox is perhaps the central charm of AI which lures many people.

It is also the core of a famous thesis by J. R. Lucas of Oxford who, citing the equivalence between programs and formal systems, invokes Gödel's Theorem to "prove" that any mechanical (i.e., computerized) model of mind inevitably is incomplete, and weaker than the human minds which create it. (It is an interesting (though perhaps irrelevant) aside to point out that contemporary computer chess and checkers programs routinely beat their programmers. Curiously enough, the best such programs resemble in very few ways advanced AI research; instead of relying on complex, flexible, self-scheduling and self-indexing programs, they rely on powerful, rigid, brute-force search techniques.) Sloman, like many other AI writers, cavalierly dismisses the Lucas thesis with a few remarks about "open" and "closed" systems. I personally am not satisfied with such cursory treatment and in my own book, *Gödel, Escher, Bach: an eternal golden braid*, I offer several alternate refutations of Lucas.

In my view, the real contribution of *The computer revolution in philosophy* is Part II: "Mechanisms". The Chapter titles are: Sketch of an Intelligent Mechanism; Intuition and Analogical Reasoning; On Learning About Numbers: Some Problems and Speculations; Perception as a Computational Process; Conclusion: AI and Philosophical Problems. (There is also an epilogue.)

In Chapter 8 (the "numbers" chapter), Sloman states bluntly (p. 181): "I shall not be talking about events or processes or mechanisms in the human brain. Exactly how the human brain works is as irrelevant to our problems as the detailed workings of a computer are to an explanation of a computer program written in a high-level programming language." This "antireductionist" position is a key thesis of the book and although it is swallowed hook, line, and sinker by a large percentage of present-day AI workers, this reviewer is not so confident. And, as a matter of fact, Sloman himself seems to be a little less positive when he states (p. 255), "When we begin to develop programs which approximate more closely to human competence, we shall have to use additional criteria [for judging the quality of AI work], including comparisons of implementation details, and of the underlying machines presupposed."

In any case, however, as Sloman repeatedly emphasizes, we are a long, long way from trying to explain the brain in all its complexity. At this stage of the game, "instead of physiological theories," he fervently argues, "we need 'computational' theories" (p. 225)–in other words, "study of the brain's *programs*, not its *architecture*" (p. 108).

Sloman begins his discussion on AI with a general sketch of a intelligent mechanism. This is a useful document setting forth in explicit terms what many AI workers implicitly know–but it is useful, because often one overlooks one aspect or another. Sloman sees ten major components:

(1) an environment,
(2) a store of factual beliefs and knowledge,
(3) a store of resources (for instance a dictionary and previously learnt procedures for making things, solving problems, etc.),
(4) a catalogue of resources,
(5) a motivational store,
(6) a process-purpose index (action-motive index),
(7) various temporary structures associated with ongoing information processing,
(8) central administrative processes,
(9) a set of monitoring processes including both permanent general-purpose monitors and others which are more specialised and are set up temporarily in accordance with current needs; and finally
(10) a retrospective analysis process, reviewing current beliefs, procedures, and plans on the basis of records of previous occurrences.

(All the above, as well as the following, is taken from p. 115.) Components 8–10 are "more permanent processes ensuring that the actions which occur are relevant to the current motives and that intelligent use is made of previous knowledge and new information. The system must have several kinds of processes running simultaneously, so that implementing it on a computer will require multi-processing time-sharing facilities–already available on many computers."

He is at pains to clarify that these components may be so intertwined that each one can be, in a sense, a 'part' of all the others. To illustrate the meaning of this, he briefly describes the concepts of recursion and mutual recursion (applicable to dynamic programs), and nested and circular lists (applicable to static data structures). These computational notions, second nature to AI workers, do indeed seem to violate common-sense notions of hierarchical organization. Despite their paradoxical feel, they are entirely consistent notions and are at the basis of most AI programs. In fact they are built into most AI languages–e.g., LISP, POP-2, and others.

How is knowledge best encoded in computer programs and data structures? This key question–the "representation question"—is the hottest issue in AI today. A major debate is the "declarative-procedural controversy": Is knowledge contained in programs, or in data structures? Sloman, along with many others, has reached the conclusion that both are necessary, and the key is interconvertability: there is a "need to blur the distinction between information-structures and programs" (p. 207). Also, on p. 201, "The distinction between data structures and programs has to be rejected in a system which can treat program steps as objects which are related to one another and can be changed."

In other words, Sloman sees *self-cataloguing* and *self-modifiability* as being at the crux of any intelligent program.

The first idea, self-cataloguing, has to do with the ways we unconsciously store information for later retrieval. For instance, we all have had the experience of remembering that a certain passage we want to find in a recently read book is on a particular part of a left-hand or right-hand page. We know that we tend to remember conversations best when we recall their entire physical contexts. These associative modes of cataloguing are not well understood by psychologists or AI workers. It is far too expensive to store explicit cross-references between all the aspects of every event we experience. Yet in cutting down on what is stored, one must be careful not to trim away so much that memories become irretrievable. Therefore, as memory size grows, one must maintain increasingly sophisticated indexing mechanisms for access to relevant data and programs. Sloman places considerable stress on the need for highly developed indexing strategies–something which is seldom pointed out in such a strong way in the AI literature.

In his chapter on how children acquire number knowledge, Sloman points out that "if an item in a structure . . . has a very long chain of associations, it might be preferable to replace the linear chain with a local index to avoid long searches. So, instead of 'three' being linked to a linear list of associations, it would have some kind of structured catalogue." (p. 210)

In fact, a two-page spread (pp. 208–209) contains an elaborate representation of extremely elementary knowledge about small integers in an efficiently cross-indexed form. Referring to this figure, Sloman says, "when the rest of the mechanism is taken for granted, a structure of the kind discussed here looks like a program for generating behaviour, but when one looks into problems of how a structure gets assembled and modified, how parts are accessed, how the different stopping conditions are applied, etc., then it looks more like an information structure used by other programs." (p. 212)

One of Sloman's main research interests is how to make a program which understands mathematics in the way a child does. On pp. 214–215, he writes:

> It will probably prove helpful to think of mathematical discovery by analogy with a program which discovers new facts about itself by a combination of executing parts of itself and examining some of its instructions. In the process it might decide that some things could be done more quickly in a different way. Or it might discover, by analysing its own structure, that instead of executing bits of programs, it can work out their effects by reasoning about them.
>
> More importantly, it may discover ways of generalizing and extending its procedures to accomplish more tasks of the same sort, or new kinds of tasks. Programmers often discover unexpected ways of elaborating and generalising their programs, in the course of examining and using them, much as an artist learns more about what he can and should do by examining an incomplete work. A program which builds its own programs can do this too. Sussman's "Hacker" program (1975) builds programs, and, in some cases, generalises them.
>
> I believe that similar ideas are to be found in Piaget's writings. Computer models turn such thoughts from vague speculations to testable theories. . . .
>
> These sorts of discoveries do not fit the normal definition of 'empirical'. For example, they need not involve the use of the senses to gain information about the world. . . . And the same is true of many other discoveries about

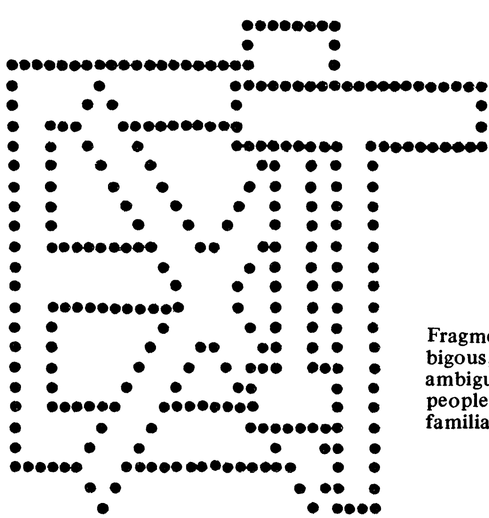
> properties of the procedures we use. Yet such mathematical discoveries involve a kind of exploration of possibilities which is closely analogous to empirical learning.
>
> We need a richer set of distinctions than philosophers normally employ. There is learning from sensory experience and learning from symbolic experience. . . .
>
> The task of designing programs which simulate these sorts of human learning to a significant extent is at the frontiers of current research in artificial intelligence. . . .
>
> The old nature-nurture (heredity-environment) controversy is transformed by this sort of enquiry. The abilities required in order to make possible the kind of learning described here, for instance the ability to construct and manipulate stored symbols, build complex networks, use them to solve problems, analyse them to discover errors, modify them, etc.,–all these abilities are more complex and impressive than what is actually learnt about numbers!

An example of a recent program in this area of AI is "AM", by Douglas Lenat. Beginning with a small amount of set-theoretic knowledge (membership, union, intersection, and so forth), and with an enormous programmatic body of heuristic knowledge about "mathematical interestingness", AM gradually developed new concepts, including: cardinality, addition of integers, multiplication of integers, factor, number of factors, minimum number of factors, prime number, sum of prime numbers, Goldbach's conjecture (every even number is the sum of two primes), maximum number of factors, "round" numbers (a notion first explored by Ramanujan and Hardy in the 1920s, and which was unknown to the programmer, Lenat). Unfortunately, it did not go on to develop any truly new notions or conjectures, but seemed to "stagnate". Lenat hypothesizes that this is because the heuristics themselves were intrinsically nonmodifiable. Which brings us back to Sloman's second idea: self-modifying programs and data structures. On this topic, he is less verbose; a typical comment (p. 201) runs like this: "these instructions need to be stored in a form which is accessible not only for execution but also for analysis and modification, like inserting new steps, deleting old ones, or perhaps modifying the order of the steps".

When one's mind has been appropriately boggled by the complexity of this sort of task, Sloman, to his credit, points out that "I cannot explain these and many more things that even primary school children learn." (p. 213) Reassuringly, he also reminds us that "children need a lot of practice at 'finding their way about' their own data-structures" (p. 210).

An issue which all AI workers have to face, but which few face with as much directness, is that of the trade-offs between explicit and implicit knowledge, between efficiency in time and storage space, between generality and specificity. "Investigations of such trade-offs between different representations is central to artificial intelligence but has hitherto been absent from philosophical discussions of rationality and most psychological theorising about cognitive processes." (p. 195) Any mode of representation when implemented in a working program leads to very definite performance characteristics. One of the most famous effects of this sort is known as the "Waltz effect", after David Waltz's vision program which, by using efficiently encoded detailed knowledge about line junctions in drawings of scenes, was able to circumvent what seemed like a certain combinatorial explosion, and which consequently outperformed its predecessors in several ways. Sometimes theories which seem before implementation to be rivals turn out to be several orders of magnitude apart from each other in running speed, simply because it is very hard to predict in advance what will happen when the many subsystems actually do interact with each other. Recently, the U.S. Government funded several AI-oriented speech-understanding projects at various sites; there were enormous differences in the performance characteristics of the different programs. The "winning" program was a dark horse called "Harpy", developed at Carnegie-Mellon University, which, like Waltz's vision program, relied heavily upon "pre-compiled networks" of knowledge. The point of this is that you just don't know how good a theory of some aspect of intelligence is, until you've tested it.

By contrast, it is the lack of this kind of concrete testability that allows many psychological theories of various mental functions to flourish simultaneously. Thus, Sloman feels, only by implementing their theories can cognitive psychologists ever hope to find out what's right and what's wrong with them. In particular the present-day debate about "Fregean" (or 'propositional') vs. "analogical" representations of visual images will only be settled by seeing which, if either, leads to successful performance by AI programs. Sloman explores this debate in depth in Chapter 7 and suggests that the two types of representation are both necessary, and complement each other.



Fragments of this picture are quite ambiguous, yet somehow they help to disambiguate one another, so that most people see a pile of letters forming a familiar word.

He is not merely drawing his conclusions out of thin air, for over the past decade, he has been involved in AI research projects– particularly vision. With his colleagues at Sussex, he has developed a program called "POPEYE" which tries to find words in dot patterns (e.g., see figure [p. 219]). How do we perceive such patterns? "Perceived fragments require a context for the interpretation. The trouble is that the context usually consists of other equally ambiguous, incomplete, or possibly even spurious fragments." (p. 218)

Sloman's theory involves a system in which many different sorts of knowledge at different levels interact, such as:

a. discerning features in the sensory array
b. deciding which features to group into significant larger units
c. deciding which features to ignore because they are a result of noise or coincidences, or irrelevant to the present task
d. deciding to separate contiguous fragments which do not really belong together
e. making inferences which go beyond what is immediately given
f. interpreting what is given, as a representation of something quite different
g. noticing and using inconsistencies in an interpretation so as to re-direct attention or re-interpret what is given
h. recognising cues which suggest that a particular mode of analysis is appropriate, or which suggest that a particular type of structure is present in the image or scene depicted

(taken from p. 220). Interestingly enough, this theory resembles in considerable detail the structure of hearsay II, the runner-up to Harpy in the speech-understanding sweepstakes.

It is rather amazing that these sorts of abilities, which we take for granted, turn out to involve stupendously large programs: (p. 13)

> A frequent discovery, using the new methodology, is that what seemed simple and easy to explain turns out to be very complex, requiring sophisticated computational resources, for instance: seeing a dot, remembering a word, learning from an example, improving through practice, recognising a familiar shape, associating two ideas, picking up a pencil.

Indeed, in the past quarter century, AI seems to have involved a constant retreat fıom ambitious plans to tackling more and more limited projects. But Sloman is convinced that we have finally achieved the modesty, the technology, and the theoretical notions to study the mind. In so doing, we will come to ask many questions that it ordinarily wouldn't even occur to us to wonder about. For example (p. 12):

> how desires and beliefs are capable of generating action . . .
> how an individual finds relevant beliefs in his huge store of information
> how conflicting motives enter into the process
> how beliefs, purposes, skills, etc. are combined in the design of an action (e.g. an utterance) suited to the current situation.

Clearly any theory of mentality ultimately has to deal in some way with the issues of emotions, consciousness, and free will. As far as emotions are concerned, Sloman seems to have a deeper respect and a deeper insight than most AI workers do. His view (like that expounded in my book *Gödel*, *Escher*, *Bach*) is that particular parts of a program or data structure are not, in isolation, responsible for emotions, but that emotions are the result of complex interactions between subprograms (p. 268):

> [I]t is important to be on guard against superficial computer models. Often by clever programming, people can produce quite convincing displays of something like a mental state, when closer inspection reveals that something very different was going on. For example, if hunger, or degree of paranoia, is represented as the value of some numerical variable then that clearly does not do justice to what are actually very much more complex states in people.
>
> . . .
>
> More complex desires, emotions, attitudes, etc., involve a large collection of beliefs, hopes, fears, thinking strategies, decision-making strategies, and

> perhaps conflicts between different sub-processes . . . At the moment, modelling such aspects of the human mind adequately is simply beyond the state of the art.

Here he is referring to programs such as Kenneth Colby's "PARRY", a slick facsimile of a paranoid obsessed with horse-racing. PARRY is not a typical AI program, for it attempts to bluff its way into credibility by shrewdly directly a conversation to its own single area of "knowledge"–horse-racing. Furthermore, its excuse for seeming so unable to converse on anything else is, conveniently, its paranoia. Sloman rightfully feels that such shallow imitations of the human psyche do no service to the name of AI.

Despite his occasional arrogant jibe, Sloman is basically a conservative AI researcher ("I do not believe that the progress of computer vision work by the end of this century will be adequate for the design of domestic robots, able to do household chores like washing dishes, changing nappies on babies, mopping up spilt milk, etc." (p. 239)), with the optimistic viewpoint that (p. 13) "a few years of programming explorations can resolve or clarify some issues which have survived centuries of disputation. Progress in philosophy (and psychology) will now come from those who take seriously the attempt to *design a person*." On pp. 239–240 Sloman presents a "disclaimer" of sorts: a list of "reasons for saying that existing computer models cannot be accepted as explaining how people do things". It is refreshing to see this kind of candid assessment of the state of AI.

For those who feel that free will and consciousness are the elusive qualities which will forever distinguish people from "mere mechanisms", Sloman has an entire final chapter dealing with such issues. The following may not convince skeptics:

> One of the sad and yet exhilarating facts most programmers soon learn is that it is hard to be sufficiently imaginative to anticipate the kinds of behaviour one's program can produce, especially when it is a complex system capable of generating millions of different kinds of processes depending on what you do with it. It is a myth that programs do just what the programmer intended them to do, especially when they are interacting with compilers, operating systems and hardware designed by someone else. The result is often behaviour that nobody planned and nobody can understand. (pp. 15–16)

A more concrete argument is this (pp. 266–267):

> A robot, like a person, could have built into it mechanisms which succeed in altering themselves beyond recognition, partly under the influence of experiences of many sorts. . . . A self-modifying program . . . interacting with many people in many situations, could develop so as to be quite unrecognisable by its initial designer(s). It could acquire not only new facts and new skills, but also new motivations; that is desires, dislikes, principles, and so on. If this is not having freedom and being responsible for one's own development and actions, then it is not at all clear what else could be desired under the name of freedom.

Of course, it is not *provable* that an AI program–or indeed a human being–has free will. Rather, it is a matter of judgment–and as to the generosity of the human race to grant its eventual mechanical thinkers "souls", Sloman is rather pessimistic:

> History suggests that the invention of such robots will be followed by their exploitation and slavery, or at the very least racial discrimination against them. Will young robots, thirsty for knowledge, be admitted to our schools and universities? Will we let them join our clubs and societies? Will we let them vote? Will they have equal employment opportunities? Probably not. Either they will be forcibly suppressed, or, perhaps worse, their minds will be designed to have limits: both their desires and their intellectual potential will be manipulated so as to safeguard the interests of people, like the 'deltas' in Huxley's *Brave New World.*
>
> It is interesting that so many people find the Brave New World techniques abhorrent when applied to human test-tube babies, but would not mind similar treatment being dealt to robots. Is it too extreme to call that racialism?
>
> . . .
>
> Where it will all lead to, we cannot foretell. My only hope is that we shall be lucky enough to produce a breed of machines with the wisdom and skill to teach us to abandon all those deep insecurities which turn us into racialists of one sort or another–probably closely connected with the processes which turn people to religion.
>
> The state of the world gives little cause for optimism. Maybe the robots will be generous and allow us to inhabit asylums and reserves, where we shall be well cared-for and permitted to harm only other human beings, with no other weapons than clubs and stones, and perhaps the occasional neutron-bomb to control the population. (p. 273)

On this cynical note he concludes his book.

However, I would like to describe some of the more intriguing hypotheses which Sloman offers, with regard to the light which AI may shed on consciousness. He suggests that consciousness is that "emergent" property of an information-processing system which arises when its central administrative process is capable of selectively focusing attention upon various subprocesses. But significantly he realizes that this conception of consciousness may well fall short of the mark (pp. 251–252):

> It is possible (as I believe Leibniz claimed) that instead of there being one division between what is and is not conscious in a complex system, there may be many divisions–one for the system as a whole, and more for various sub-systems. If there is something in the argument about the need for some centralised decision-making in the system as a whole, then the same argument can be used for the more complex sub-systems: considered as an organic whole there may be some things a sub-system can be said to be conscious of, and others which it cannot.
>
> . . .
>
> Maybe that is the best way to think of a person: but if so we shall not fully understand why until our attempts to design a working person have forced such organisations on us.

Indeed this is the whole thrust of Sloman's book. He is vitally concerned with conveying the excitement and challenges of AI. If he is a crusader, it is because he has personally witnessed the power of AI and is concerned with retrieving his fellow philosophers from drowning in abstruse theoretical verbiage which slowly becomes less and less relevant. Indeed it is safe to say that Sloman regards AI as a new synthesis of science and philosophy: *experimental philosophy*. He offers a challenge to those who bitterly attack AI:

> Anyone who objects to a particular explanation in the form of a program, should try to construct another better explanation of possibilities, that is, better according to the criteria by which explanations are assessed. . . . The preferred explanation should account for at least the same range of possibilities with at least as much fine structure. (p. 111)

Without insisting that it be a program, he does demand an equally complete explanation from any rival theory. Those who criticize AI should ponder this well.

If Sloman's book has the impact he hopes, it will certainly create what its title proclaims: a computer revolution in philosophy.

I have a few gripes with the way the book as a whole is put together: (1) it is riddled with typos and bad punctuation which do not impair understanding but which lower one's estimate for the amount the author cares about his work; (2) occasionally long passages appear in boldface, or reduced, or indented, for no apparent reason; (3) too many brief asides are thrown in for some special restricted audience, and they detract from the flow; (4) its tone is simply too biting.

But despite all my reservations, Sloman's book is a significant and highly original contribution to the debate about minds and machines.

DOUGLAS R. HOFSTADTER

BULLETIN (New Series) OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 2, Number 2, March 1980

0002-9904/80/0000-0110/$02.75

Interpolation theory, function spaces, differential operators, by Hans Triebel, North-Holland Publishing Company, Amsterdam, New York, Oxford, 1978, 528 pp., $66.75.

The first interpolation theorem was given by M. Riesz (1926) in his study of the L_p mapping properties of certain operators associated with the Fourier series. Riesz showed that the boundedness of a linear operator A as a mapping from $L_{p_i}(\mathbf{T})$ to $L_{r_i}(\mathbf{T})$ ($i = 0, 1$) carries with it the boundedness of A from $L_p(\mathbf{T})$ to $L_r(\mathbf{T})$ for other pairs (p, r). The power of the method is that it determines the mapping properties of A on L_p spaces by examining A on only two appropriate pairs of (endpoint) spaces.

Much of the early work in interpolation centered around extending and refining Riesz's results to be applicable to a larger variety of operators. It was not until the development of the abstract methods of interpolation in the late 1950s that the wide applicability of interpolation became clear. These abstract methods not only allow for the study of operators on general Banach spaces but also give a unified approach to the development of various classical families of spaces which arise in the modern theory of differential equations, approximation, and numerical analysis. With the development of these abstract methods, interpolation has become a major discipline which is indispensible for a thorough understanding of that portion of analysis which deals with spaces of functions and mappings of operators.