CHAPTER 5

ARE COMPUTERS REALLY RELEVANT?

Experience has shown that many readers will have been made very uncomfortable, if not positively antagonistic, by my remarks about the role of computing and computer programs in philosophy and the scientific study of human possibilities. There are several reasons for this, including (a) ignorance of the nature of computers and computer programs, (b) misunderstandings about the way computers are used in this sort of enterprise, (c) invalid inferences from the premises that computer simulations of human minds are possible, and (d) confused objections to specific theories expressed as computer simulations.

5.1. What is a computer?

It is not helpful to think of a computer simply as something which does numerical calculations, for this is only one use of a far more general facility. A computer is a mechanism which interacts with symbols. It can accept symbols, store them, modify them, examine them, compare them, construct them, interpret them, obey them (if they express instructions), or transmit them. It must therefore include a 'store' or 'memory' containing a large number of locations at which symbols can be stored. These locations must be 'addressable': that is, it should be possible for an instruction somehow to mention a location so that its contents can be examined or something new put there. The mechanism may assume that all the basic symbols stored used some fixed format, such as sequences of zeros and ones, but that is no restriction, as sufficiently complex combinations of such symbols can be used to represent anything, just as complex sequences of the simple characters on a typewriter can express poems, plays, propaganda or physical theories, or complex arrays of dots can be seen as photographs of faces.

Since the symbols stored in the computer may include instructions for it to obey, and since it can be instructed to change some or all of the symbols within it, it follows that as a computer executes instructions within itself, the instructions may change and thus the processes occurring may evolve in complex ways. In the end, the original program may have completely disappeared. Exactly how this happens may depend not only on the original program but also on the history of interactions with the environment. So no programmer, or anybody else, is responsible for the eventual state of such a mechanism or for its behaviour.

In any modern digital computer the basic symbolic processes which occur will all be very simple, such as putting a zero or a 1 in some location, or comparing two symbol-strings, or copying the contents of one location into another, or performing logical or arithmetical operations. But it is not helpful to think of a computer as 'simply' performing such simple operations, any more than it is helpful to think of a Shakespeare play as 'simply' composed of letters, punctuation marks, and spaces.
Computers can perform millions of their basic operations each second. Many different kinds of books can be written using the same small set of printed characters, and similarly an enormous variety of processes can be represented by complex combinations of the simple processes in a computer.

In particular, the processes need not be fully controlled by all the symbols in the store at any time. For among the instructions executed may be some to the effect that new symbolic information should be accepted from various devices attached to the computer, such as a television camera or a microphone, or a teletype at which a person sits communicating with the computer. Some of the new symbols coming into the computer in this way may lead to changes in the stored instructions, just as much as execution of stored instructions can. (This, incidentally, is why all the philosophical debates about Godel’s incompleteness theorem and related theorems proving that there are limits to what any particular computing system can do, are irrelevant to the problem of what sorts of intelligent mechanisms can be designed: for all these theorems are relevant only to ’closed’ systems, i.e. systems without means of communicating with teachers, etc.)

Computing science is still in a very early phase. Only a tiny fragment of the possible range of computer programs has so far been investigated, and many of these are still only partly understood. 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 text-books 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).

The last general remark I wish to make about computers is that the definition given above does not assume anything about what the mechanism is made of. It could be transistors, it could be more old-fashioned electronic components, it could be made of physical components not yet designed, it could somehow be made out of some non-physical spiritual stuff, if there is any such thing. The medium or material used is immaterial! All that matters is that enough structures are available to represent the required range of symbols, and that appropriate structural changes can occur in the computer. As Margaret Boden once remarked, angels jumping on and off pin-heads would do.

This is not the place to enlarge further on what computers are. Interested readers should consult Electronic Computers, by Hollingdale and Toothill, or Weizenbaum’s Computer Power and Human Reason. See also chapter 8 of this book.

5.2. A misunderstanding about the use of computers

I have heard people talk as if computers were some new kind of organism, distantly related to humans or other animals, so that one might perhaps learn something about animals or their brains by studying computers!

However, computers are not natural objects to be studied. They are artefacts to be improved and used. If people had been content to study computers instead of programming them, very little would have been learnt, for a computer does nothing unless it is programmed. But what it does depends on how it is programmed. So approaching a computer with a view to finding out what it can do is as silly as it would be for a physicist to study pencil and paper with a view to finding out what they can do. One approaches a computer in order to try to make it do something. The physicist writes things down,
calculates, tries out formulae and diagrams, etc. He constructs, explores and modifies a theory. That is how to use a computer in order to study intelligence: by designing a program which will make it behave intelligently one constructs a theory, expressed in that program, about the possibility of intelligence. The failure of the theory is your own failure, not the computer’s.

So objections to the discipline of artificial intelligence based on the assumption that its practitioners study computers are completely misguided.

5.3. Connections with materialist or physicalist theories of mind

Many readers (some sympathetic and some unsympathetic) will jump from the premiss that computer programs can simulate aspects of mind, or can themselves be intelligent and conscious, to the conclusion that some kind of materialist or physicalist theory of mind is correct. Alternatively they will assume that because I stress the importance of computing studies, I support some kind of reductive materialist theory. There are two answers to this.

The short answer is that just because an electronic computer is a physical system, it does not follow that everything it successfully simulates is a physical system: there could be computer programs simulating the structures and functions of mechanisms composed of some spiritual substance!

So even if the human mind is not merely a function of the physical brain, but has some non-material or non-physical basis (whatever that may mean), then the behaviour or function of that stuff is what computer programs can simulate. In fact a program does not specify what kind of computer it runs on. The computer may use transistors, valves or spiritual mechanisms, so long as a rich enough variety of structural changes is available, as I have already pointed out.

A longer, and more important, answer is that the ontological status of mind has little relevance to the problems of this book. Both Dualism, which postulates some kind of spiritual entity distinct from physical bodies, and Materialism, according to which minds are just aspects of complex physical systems, lack explanatory power. That is, both of them fail on the criteria proposed in chapter 2 for adequate explanations in philosophy or science. They fail either to describe or to explain any of the fine structure of such aspects of mind as perception, memory, reasoning, understanding, deciding, desiring, enjoying, creativity, etc., or the relations between them.

In order to explain how all these things are possible, we need a theory describing or representing the structures and functions of a mechanism which can be shown to have the right sorts of abilities, that is a mechanism able to generate within itself structures and processes with the kinds of mutual relationships which we know hold between mental phenomena. For instance, we know that a certain experience, such as seeing a tool being used, can produce a change in what a person knows, and thereby can change what he is able to do and the decisions he can take in order to deal intelligently with problems. To explain how this sort of thing is possible, e.g. to explain how one can learn to operate a tool by watching its use, it will not do simply to say what kind of stuff the underlying human mental mechanism is made of.

Being told that a computer is made of physical components, for instance, tells you nothing about the kind of internal organisation that made it possible for the PDP-IO computer used by Winograd (1973) to hold conversations in ordinary English. Similarly, being told that the mind is spiritual or non-physical explains nothing.
For similar reasons, neurophysiology cannot help in the early stages of the search for explanations of the possibility of mental phenomena and we shall remain in the early stages for some time. Studies of neurophysiology, or the electronic basis of a computer, may explain such things as how fast the system performs, and why it sometimes goes more slowly, or why it sometimes breaks down altogether; but cannot at present explain how it is possible for the system to perform a particular type of task at all. Such an explanation requires study of the brain’s programs, not its low level (physical) architecture and neurophysiology currently lacks conceptual and other tools needed for studying programs. (Study of a computer’s architecture tells one practically nothing about the programs currently running on it. The programs may change drastically while the physical architecture remains the same, and different computer architectures may support the same programs. Computers are not like clocks.)

[Note added 2001: I would now put this by saying that the virtual machine architecture is more important than the physical machine architecture. (For more on this see recent papers in http://www.cs.bham.ac.uk/research/cogaff/). The study of physical architectures would be relevant if could be used to demonstrate that certain sorts of virtual machines could and others could not run on brains. But right now we still do not know enough about ways of mapping virtual machines onto physical machines for useful constraints to be derived. ]

The only kinds of explanatory mechanisms that have some hope of being relevant to explaining mental possibilities like perception, learning and decision making, are mechanisms for manipulating complex symbols, for example, computer programs.

People whose sole experience of computing is with programs for doing highly repetitive algorithmic numerical calculations, or programs for simulating feedback systems, may find it hard to understand how programs can be relevant to our problems. An essential antidote to this prejudice is a study of the literature of artificial intelligence to learn how, besides doing numerical calculations in an order determined by the programmer, computer programs can also construct, analyse, interpret, manipulate, and use complex symbolic structures, like lists, pictures, sentences or even sub-programs, in a flexible way determined by analysis of developments during the computation rather than following an order worked out in advance by the programmer.

All this can be summarised by saying that the known important mechanisms are not computers (those ugly boxes with mysterious noises and flashing lights), but programs or virtual machines. Computers are an old type of mechanism: they are physical machines. Programs are a new type. A simulation program could drive not only a physical computer, but, if ever one were made, a computer composed entirely of spiritual stuff (The program, not the medium, is the message.)

**5.4. On doing things the same way**

The persistent objector may now argue that the explanatory power of computer programs is doubtful, since even if a program does give a machine the ability to do something we can do, like understand and talk English, or describe pictures, that leaves open the question whether it does so in the same way as we do; so it remains unclear whether the program gives a correct explanation of our ability.

The objector may add that it is clear that existing computers do not do things the way we do, since, at the physical level they use transistors and bits of wire, etc., whereas our brains do not, and even at the level of programs they have to employ interpreters or compilers which translate the high level intelligent and flexible symbol-manipulating programs into sequences of very simple and very mechanical instructions which have to be followed blindly, whereas there is no evidence that humans do this.
This objection (which seems to pervade the book by H.L. Dreyfus, *What Computers Can’t Do*), is based on the concept ‘doing things in the same way’, which requires some analysis.

The notion of doing something *in the same way* is systematically ambiguous. Two persons may calculate the answer to an arithmetical question in the same way insofar as they both use logarithms but in different ways insofar as they use logarithms with different bases. It is all a matter of how much and what sort of detail of a process is described in answer to the question In what way did he do it?’ That some very detailed description would be different in the case of a computer does not imply that there is no important level at which it does something the same way as we do. We don’t say a Chinaman plays chess in a different way from an Englishman, *simply* because he learns and applies the rules using a different language, so that his thinking goes through different symbolic processes. He may nevertheless use the same strategies.

The same problem arises about whether two computer programs producing equivalent results do so in the same way. Two programs using essentially the same algorithm may look very different, because they are written in different languages or in different programming styles. Any program is a mixture of main ideas and implementation details. The same may be true of human abilities.

The problem of knowing the way in which a computer does something is no different in principle from the problem of knowing the way in which a person does it. In both cases there are questions that can be asked, and tests that can be given, which provide useful clues. (Compare Wertheimer’s tests for whether children understand and apply a technique for finding areas of a parallelogram in the same way as he does, in *Productive Thinking*, chapter I. He sees whether they can solve a very varied range of problems.)

Insofar as anything clear and precise can be said about ‘the way’ in which a human being does something (e.g. plays chess, interprets a poem, or solves a problem) the appropriate procedure can in principle be built into a suitable simulation, so that we ensure that the machine does it in the same way. For instance, programs can be written to do multiplications using ordinary decimal arithmetic, or binary arithmetic, or alternatively using natural language.

Finally it should be noted that it is very unlikely that there is only one way in which something or other is done by all human beings, whether it be perceiving faces, remembering names, playing chess, solving problems, or understanding a particular bit of English: we all have our own quirks and foibles, so it is unreasonable to deny this right to a complex computer simulation.

I do not wish to argue that every aspect of the human mind can be simulated on digital electronic computers, any more than an astronomer’s explanation of an eclipse explains or predicts every aspect of the motion of the earth, moon and sun. For instance, certain types of human experience seem to be possible only for beings with human bodies, or bodies with very similar structures. Thus, feeling thirst, nausea, muscular exhaustion, sexual desire, the urge to dance while listening to music, or the complex combination or bodily sensations when one is about to lose one’s balance whilst walking on ice, may be forever inaccessible to computer programs within immobile rectangular boxes, or even to humanoid mobile robots who are made mainly of plastic and metal. (For more on these general issues, see the contributions by H.L. Dreyfus, N.S. Sutherland, and myself to *Philosophy of Psychology*, ed. S.C. Brown.)

These abstract debates about what can and cannot be done with computer programs are not too important. Usually there is more prejudice and rhetoric than analysis or argument on both sides. What is important is to get on with the job of specifying what sorts of things are possible for human minds, and trying to construct, test, and improve explanations of those possibilities. Anyone who objects to a particular explanation expressed in the form of a program, should try to construct another better
explanation of the same range of possibilities, that is, better according to the criteria by which
explanations are assessed (see chapter 2). The preferred explanation should account for at least the
same range of possibilities with at least as much fine structure.

The rest of this book will be concerned mainly with the description of some important possibilities
known to common sense, together with some rather sketchy accounts of what good explanations might
look like. I shall frequently point out ways in which the attempt to design computer simulations can
subserve the endeavour to understand the human mind.

[[Note added 2001:
After this book was published there was a revival of interest among many AI researchers in
"connectionist" architectures. Some went so far as to claim that previous approaches to AI had
failed, and that connectionism was the only hope for AI. Since then there have been other swings
of fashion. It should be clear to people whose primary objective is to understand the problems
rather than to win media debates or do well in competitions for funding that there is much that we
do not understand about what sorts of architectures are possible and what their scope and
limitations are. It seems very likely that very different sorts of mechanisms need to be combined
in order to achieve the full range of human capabilities, including controlling digestion,
maintaining balance while walking, recognising faces, gossiping at the garden gate, composing
poems and symphonies, solving differential equations, and developing computer programs such
as operating systems and compilers. I don’t know of an any example of an AI system, whether
implemented using neural nets, logical mechanisms, dynamical systems, evolutionary
mechanisms, or anything else, that is capable of most of the things humans can do including those
items listed above. This does not mean it is impossible. It only means that AI researchers need
some humility when they propose mechanisms. ]]

[[Note added 20 Jan 2002:
A number of arguments against computational theories of mind have been advanced since this
book was written. Many of them use arguments that were already rebutted in this chapter, or put
forward views that were expressed in this chapter. For example, the argument that brains work in
different ways from computers therefore computational theories of mind must be incorrect is
rebutted above by pointing out that systems may be different at one level of description and the
same at a more abstract level of description. Abstraction is often very useful, as demonstrated by
the history of science in general and physics in particular. The argument that intelligence or
mentality requires embodiment is rebutted by pointing out that some aspects of mind may depend
on details of the body whilst others do not. Of course, that leaves unanswered the important
research question: which forms of embodiment can support which forms of mentality?

Many critics of AI and some defenders of AI have based their argument on the assumption that
AI in some sense presupposes that all computation is Turing Machine computation. I have tried to
argue in recent years that the notion of "computation" is not sufficiently well defined to support
such criticisms. In particular I have argued that the notion of "computation" employed by most
users of computers, designers of computers, programmers, and AI researchers, has nothing to do
with Turing machines but is an extension of two notions which go back to long before Turing, namely

a) The notion of a machine that can control something, possibly itself
b) The notion of a machine that operates on abstract entities, such as numbers, or census
information.

Both ideas were well advanced before the beginning of the twentieth century, for instance in
automated looms, mechanical calculators and Hollerith machines for sorting and collating
information. In the middle of that century advances in science and technology made it possible to combine those ideas in new ways, providing far greater speed, power, flexibility (e.g. self programming), and cheapness. These points are elaborated in a paper on the irrelevance of Turing machines to be published during 2002, and other papers available here:
http://www.cs.bham.ac.uk/research/cogaff/

Despite all the progress of the last half century, it is clear that we still have much to learn about the nature of information and varieties of machines, including virtual machines, that can process information -- themes developed in these talks:
http://www.cs.bham.ac.uk/research/projects/cogaff/talks/ and these discussion papers
http://www.cs.bham.ac.uk/research/projects/cogaff/misc/}