FIRST PROOF

THE TIMES HIGHER OCTOBER 22 2004

Tony Hoare disputes Eric B. Baum's hard-held view that the

All in the mind – but not in

What is thought? A short answer can be found from sentences in the last chapter of this long book: "The mind is an evolved computer program", "Thought is the execution of this computer code", "There is a similar explanation for evolution itself... life and the biological evolution that ultimately produced mind are simply particular types of Turing machine program."

To illustrate and support its views, the book explains many important and interesting concepts and results from many branches of science: genetics, experimental psychology, probability theory, economics, linguistics. All of these are garnished with a light sprinkling of philosophy. Occam's razor, for example, is fundamental to the minimal encodings that are the basis of attempts at machine learning. Many interesting analogies are drawn, that span several branches of science. For example, chapter two describes the genetic concept of pleitropy, by which genes are re-used on different occasions during development of an organism to perform radically differing functions. This is like the linguistic phenomenon of metaphor, by which the same word can be applied to widely differing phenomena in different contexts.

For many readers, the value of the book derives from the science that it describes in passing, even if the claims of connection between the science and the central theses of the book are hard to follow (or, for some readers, even harder to swallow).

What is Thought?

By Eric B.Baum

The MIT Press
478 pp, £25.95
ISBN 0 262 02548 5

For the author, the most relevant branch of science is computer science, from which many philosophically interesting ideas are described. Firstly, Turing's archetype of computation, the Turing machine. Equally fundamental is John von Neumann's 1948 construction of self-reproducing automata. Other relevant computational paradigms include neural nets, post-production systems, and the programming language LISP. There is a good description of the travelling salesman problem, and of computational complexity.

The author's own experiment in evolutionary programming is crucial to the thesis of the book. It aims to show that the mechanisms of evolution and natural selection are capable of giving rise to intelligent agents. First, a simple model of intelligence is taken to be just the possession of a program that can build a column of children's building blocks to match a given sequence of colours. Only one block can be moved at a time, from the top of one of three columns to the top of one of the other two columns. The starting configuration of the columns of blocks is random. As soon as the problem is explained, any programmer could see how to program an efficient algorithm. But evolution does not even have the problem explained: evolution has to learn the problem at the same time as the solution.

The first attempt to evolve the intelligent program followed the analogy of genetics. A population of programs that simulate agents was generated at random. Each was run on 20 simple examples, with only a few blocks. The less successful programs were killed off and successful programs were "mated" to produce offspring similar to themselves. Each offspring contained a mixture of parts derived from its two parents, and occasionally a random mutation. The process was repeated on a computer over millions of generations, on problems of increasing complexity. The experiment was a complete failure. Nothing like an intelligent program evolved. In fact, any signs of intelligence were systematically bred out of the population.

The second experiment followed the analogy of economics. Again a population of 1,000 simulated agents were generated, but now they would cooperate and compete to achieve the goal. Each had a balance of simulated money. One of agents owned the right to make the next move, but could auction that right to the highest bidder, who could auction it again after the move, hopefully for more than it was bought for. The agent making the final move that solved the problem received a large payout. At each generation, bankrupt agents were removed, and successful ones were replicated with occasional mutation. After a few hundred thousand training examples, the community of agents evolved to solve problems with hundreds of blocks. Each of them had learnt a different program to perform a potentially useful move. And each of them had learnt to evaluate the current situation, to decide how much to bid for the right to make the next move. Interestingly, the agents were clearly partitioned into three distinct species.

The author confesses that the experiment was partly rigged, by giving the evolving programs a jumpstart. They had access to a built-in function that told them how many blocks were already in the correct position, and this information is crucial in determining both the next move and a sensible size for the bid. When the same experiment was run on the problem of Rubik's cube, without providing an evaluation function, the results were unimpressive. But the author is not

FIRST

mind is like a computer program

the software

discouraged. "Evolution had vastly more computational resources than we do." Furthermore, evolution favours techniques that make evolution itself more effective, a process known as recursive improvement. "[As a result] evolution naturally discovers programs that interact with the environment and learn".

The book ends on an optimistic note: "Better methods for evolving programs [running on real computers] that recurse their growing intelligence to improve the evolution, coupled with additional approaches for jumpstarting the evolution and with further gains from Moore's law [that computers continue to get faster], could well produce thinking machines."

The author is an unashamed proponent of Turing's original views on the nature of mind — "strong Artificial Intelligence". He even claims: "The overwhelming consensus of the field is that the mind must be equivalent to a computer program." Not so. There are many computer scientists and others who have severe doubts on this score. They point out the many radical differences between the known structure and mechanisms of the brain and those of computers like the Turing machine. Such a machine has a single locus of control, whereas the brain is clearly distributed, with many billions of interacting components. Computers are essentially digital and discrete, whereas the brain is largely analogue and continuous. Computers are deterministic, whereas it is likely that probability has a role in explaining the working of the brain. Programs are algorithmic, and use strategy to calculate results. The brain seems to match patterns in some kind of massive table look-up of stored memories. In fact, almost everything we know about the brain accentuates its differences from a computer. Of course, computer programs can simulate analogue processes, including distribution, probability and pattern matching, just as programs routinely simulate the motions of the planets. But that does not mean that the solar system is a computer or a computer program.

The parts of the book that describe established technical ideas and results of science and maths are well written and worth reading. Surprisingly, the parts that put forward the author's own views are not of the same quality. There are too many rambling sentences that have to be read twice or more. And, even then, they often seem to be saying the same thing as the preceding sentence. Even the summaries of the sections and chapters are little better than repetition. Could this be a symptom of the yawning gap between the quoted results of established science and the conclusions that the author wishes to justify by them? If there is no logical connection that can be drawn, mere repetition has to suffice.

There are, surely, better books on the relationship between computers and the mind, but few of them espouse so wholeheartedly the cause of strong Artificial Intelligence. It is a shame that in its discussion of the crucial issues the book comprehensively fails to make its point.

Sir Tony Hoare is senior researcher, Microsoft Research, and was a student of philosophy at Oxford University.