Whoever Said Computers Would Be Intelligent?

John Self

Drakkar Press

Whoever Said Computers Would Be Intelligent?

John Self

First published in Great Britain in 2005 by
Drakkar Press Limited,
PO Box 724, Lancaster, LA2 9WZ, England

Printed by Cyclic Digital Print Ltd.,
Bridge End Business Park, Milnthorpe, Cumbria LA7 7RH, England

This version placed on-line by Drakkar Press in 2015

ISBN: 978-0-9548605-0-9

Drakkar Press, Lancaster, England

Whoever Said Computers Would Be Intelligent?

Contents

Preface

This book relies greatly on the words of others. It seems only fair therefore that I should offer some words of my own for this preface. So, I handed the book to my alter egos to review, giving them the task, for which they are eminently suited, of providing some penetrating self-criticism. Here are some of the things they said:

If you've ever picked up a book and thought that the beginning-of-chapter quotes were the best thing in it, then you'll love this book. There are beginning-of-chapter quotes everywhere, except at the beginning of chapters.

There are two kinds of book on artificial intelligence: serious, formal texts for the active researcher, unreadable by anyone else; and polemics, in which the author has to take an extreme position. Now there is a third. This is a light-hearted introduction to the field, which touches upon modern research, without technical detail, and gives a balanced view of the various debates.

This book unfolds like Sibelius's seventh symphony – a single movement in which themes are entwined and elaborated to reveal an integrated, vast vista of revolutionary ideas. A continuous, coherent narrative about AI is quite an achievement, for it often seems an assortment of isolated research fields.

Over 500 quotations are woven seamlessly into a fascinating saga taking us from Pascal to space probes. There are 66 biscuit-sized sections, each digestible during a coffee break.

The conceit of constructing a history and review of AI around the quotes of those concerned, and not so concerned, works excellently. It puts the AI programme in its social and philosophical context, by showing that its dilemmas have been the subject of debate for millennia. And it holds AI researchers to account for what they have said about AI in the last fifty years.

As an AI researcher, I am appalled at the implications of this book. We have worked very hard to develop a tradition whereby we are encouraged to prognosticate on the future of our discipline, without too much concern

for the realities of life. Now we will have to be careful what we say and write in case this miserable scribe is taking notes to hold against us.

It beats me why anyone should read this book. It leads inexorably towards predictions that humanity will be obsolete in twenty years. If this is right, why waste valuable time reading about it? If this is wrong, why waste valuable time reading about it?

At last, here is a book on artificial intelligence that your grandmother can read – and with enjoyment and enlightenment. If she keeps pestering you about what you get up to at the university and says she can't get past chapter 1 of Russell & Norvig, Nilsson, Moravec, Ginsburg or even Penrose, then give her this to keep her quiet.

Like any reviewer, I consulted the name index first. I was missing – but I was delighted to find that I would have been in the company of Michael Scriven, John Searle, Oliver Selfridge, Otta Selz, William Shakespeare, Claude Shannon, George Bernard Shaw, and many others. Who could fail to enjoy such eclecticism? I feel inspired to try to say something profound for the next edition.

Whoever Said Computers Would Be Intelligent?

1. Calculating machines: "something extraordinary"

This book is about 'computer intelligence', in quotes. It weaves together a set of remarks, pertinent and impertinent, on the subject of artificial intelligence (AI). The development of machinery that possesses some of the characteristics of intelligence will undoubtedly come to be seen as one of the most significant events of the late twentieth century. But just how significant will AI prove to be? Edward Fredkin, then director of the Laboratory of Computer Science at the Massachusetts Institute of Technology, one of the leading AI research centres, was in no doubt:

> **There are three events of equal importance ... Event one is the creation of the universe ... Event two is the appearance of life ... And, third, there's the appearance of artificial intelligence ... There can't be anything of more consequence to happen on this planet.**
>
> *Edward Fredkin (1979), quoted in Pamela McCorduck, Machines Who Think, New York: W.H. Freeman.*

So, the development of AI may be the **last** event of consequence – if it were to happen. If it has not already proven too late, we ought perhaps to form an opinion about AI.

It seems obligatory to begin with a definition of the term 'artificial intelligence' but alas there is no consensus on what precisely it means, as illustrated by the following definitions from three textbooks:

> **[AI is] the study of how to make computers do things at which, at the moment, people are better.**
>
> *Elaine Rich and Kevin Knight (1991), Artificial Intelligence, New York: McGraw-Hill.*
>
> **[AI is] the study of mental faculties through the use of computational models.**
>
> *Eugene Charniak and Drew McDermott (1985), Introduction to Artificial Intelligence, Reading, Mass.: Addison-Wesley.*
>
> **AI is the study of agents that exist in an environment and perceive and act.**
>
> *Stuart Russell and Peter Norvig (1995), Artificial Intelligence: A Modern Approach, Englewood Cliffs, N.J.: Prentice-Hall.*

These definitions immediately identify some key points of controversy: Does AI necessarily involve comparisons between humans and computers? Is AI primarily concerned with 'doing' rather than 'thinking'? And with all this

mention of intelligence and agents, is AI about espionage? Rather than embark on an abstract discussion of these issues at this stage, we will, with the help of many commentators, present a review of the development of AI up to the present and then speculate upon its (and our) future.

There seems to be something inherently contradictory in the juxtaposition of the words 'machine' and 'intelligence':

> **machine** *n vt,* **1. apparatus for applying mechanical power .. 2. person who acts mechanically and without intelligence.**
>
> *Concise Oxford Dictionary.*

However, not all man-made devices are for applying mechanical power. Are the telescope, the sundial and the abacus considered to be machines?:

> **The abacus is a hand-operated calculating machine ...**
>
> *Donald Clarke, ed. (1982), The Encyclopedia of How it Works, London: Cavendish.*

The abacus is thought to have been developed in the Orient over five thousand years ago but its use in many widely separated cultures (China, Egypt, Greece, Mexico, Peru and so on) might indicate that it was independently invented in several places. Numbers are represented by beads strung on rods or wires set in a rectangular frame and calculations are carried out by sliding the beads along the wires. The abacus is basically for adding and subtracting and there is no one-stage process for multiplication. This was, of course, a difficulty for early arithmeticians, who had numeral systems that did not allow them to write numbers in columns and so to perform multiplication on paper the way we did until recently.

The abacus does, however, lack one important characteristic that we associate with machines – it does not possess the ability to perform apparently autonomous actions. The calculations are performed by **the user** moving the beads: the beads never move 'of their own accord'. Consider the difference between a 'gun' and a 'machine-gun'. A gun fires a single bullet when we pull the trigger but a machine-gun fires long bursts, that is, one action by the user sets in train a series of actions carried out by the machine. We set a prototypical machine into action by pulling a lever or pressing a button and thereafter it performs the actions it has been designed to perform. So a 'calculating machine' designed to perform addition would be one which, after we set it off, would autonomously carry out a sequence of actions through which it could determine the result of adding the two numbers we had previously specified.

The French mathematician and philosopher, Blaise Pascal, modestly introduced such a device in 1642:

I submit to the public a small machine of my own invention by means of which alone you may, without effort, perform all the operations of arithmetic, and may be relieved of the work which has often times fatigued your spirit.

Blaise Pascal (1642).

The invention of the automatic calculator was for a long time attributed to Pascal but it is now known that a presumably even more modest German named Wilhelm Schickard, professor of astronomy at Tübingen, had designed and built such a machine in 1623. These devices could only perform addition and subtraction but nonetheless they created quite an impression, particularly with those such as astronomers, engineers and accountants whose livelihoods depended on making numerous tedious and detailed calculations. Others, however, were disconcerted:

The mind had somehow been taken over by the machine ... it could do any kind of calculation without error, which was something extraordinary to be able to do without a pen, but even more so without even knowing arithmetic.

Pascal's sister (regrettably unnamed), quoted in L. Perrier (1963), Gilberte Pascal, Bibliographie de Pascal.

This is probably the first ever comment on the subject of AI.

Over the next two centuries the technology of calculating machines was considerably refined, most notably through the introduction of multiplication and division by the German philosopher Gottfried Wilhelm Leibniz (1646-1716). It is difficult today, in the age of cheap calculators, to appreciate what a bane something like multiplication must have been but a diary entry of Samuel Pepys (1633-1703) gives an idea:

... By and by comes Mr Cooper, of whom I intend to learn mathematiques, and do begin with him today, he being a very able man. After an hour's being with him at arithmetique (my first attempt being to learn the multiplication table); then we parted until tomorrow.

Samuel Pepys (1662), Diary, July 4.

At that time, Pepys, who was as well educated as anyone of the time, was in charge of the Contracts Division of the Admiralty – and yet he needed expert tuition on multiplication.

The advent of machines capable of carrying out arithmetical operations had two outcomes. On the one hand, awe-ful ignorance provoked extremes of exaggeration, such as this description of a machine displayed at the Paris Exhibition of 1857:

This machine, among the most ingenious, solves equations of the fourth degree and of even higher orders ... scientists who vaunt their calculating

> **powers, as a divination of the laws of nature, will be advantageously replaced by a simple machine, which, under the nearly blind drive of an ordinary man, of a kind of movement, will penetrate space more surely and profoundly than they. Any man knowing how to formulate a problem and having the machine of the Messieurs Scheutz at his disposal for solving it will replace the need for the Archimedes, the Newtons or the Laplaces. And observe how in the sciences and arts, all is held together and intertwined: this nearly intelligent machine not only effects in seconds calculations which would demand an hour; it prints the results that it obtains, adding the merit of neat calligraphy to the merit of calculation without possible error.**
>
> *Baron L. Brisse (1857), Album de l'exposition universelle.*

On the other hand, "nearly intelligent" machines did not impress the realists:

> **That arithmetic is the basest of all mental activities is proved by the fact that it is the only one that can be accomplished by a machine.**
>
> *Arthur Schopenhauer.*

These two reactions we will see repeated throughout the history of AI. By the way, Schopenhauer (1788-1860) was a gloomy German metaphysician who was happy to introduce to the West various Eastern philosophies that supported his insistence on the universality of suffering. He wrote abusive texts *On Women* and also *On University Philosophy*, the latter after he had deliberately timed his lectures to coincide with those of Georg Hegel – and had then found that students preferred Hegel. His comments on arithmetic are mild in comparison.

2. Analytical Engine: "an out-of-pocket machine"

In the early nineteenth century, only the mathematician and philosopher Charles Babbage, who did much to further the tradition of British eccentricity, and his accomplice, Ada Lovelace, seemed fully to appreciate that profound ideas were taking shape. Although Babbage is recognised today only as the father of the modern computer, he was, as was the custom of the time, a polymath with views on all kinds of issues outside his official domain of expertise. For example, he wrote *On the Economy of Machinery and Manufactures* to propose a system of co-partnership for workers so that they may share in profits and join owners in decision-making – this at a time (1834) when the ruling classes were seeking to starve strikers into submission and ban trade unions by law. He also proposed the creation of an electric grid (which did not happen for another fifty years) and he recommended the decimalisation of the British currency (which eventually occurred in 1971)

although the latter is not so insightful as the French had converted to decimal currency in 1799, abandoning the 12 deniers in a sol, 20 sols in a livre system that had complicated Pascal's calculator.

Babbage first dreamed of constructing an automatic calculating machine in 1812 or thereabouts and pursued this dream relentlessly until he died in 1871, having succeeded in principle but failed miserably in practice. His first design was for a Difference Engine, intended to calculate tables of logarithms. It was based upon the 'method of differences', which is basically a way of repeatedly calculating successive values using only addition. Detailed plans and a prototype were eventually developed and submitted to the British Government for support and duly received the qualified recommendation typical of governmental advisors and politicians out of their technological depth:

My dear Peel,

> **Mr. Babbage's invention is at first sight incredible, but if you will recollect those little numerical locks which one has seen in France, in which a series of numbers are written on a succession of wheels, you will have some idea of the first principles of this machine, which is very curious and ingenious, and which not only will calculate all regular series, but also arranges the types for printing all the figures. At present indeed it is a matter more of curiosity than use, and I believe some good judges doubt whether it can ever be of any. But when I consider what has been already done by what were called Napier's bones and Gunter's scale, and the infinite and undiscovered variety of what may be called the *mechanical powers* of numbers, I cannot but admit the possibility, nay the probability, that important consequences may be ultimately derived from Mr. Babbage's principle.**
>
> *John Wilson Croker (March 21 1823), letter to Mr. Peel (Home Secretary in the British Government).*

It appears that Mr. Babbage has displayed great talents and ingenuity in the construction of his machine for computation which the Committee think fully adequate to the attainment of the objects proposed by the inventor, and that they consider Mr. Babbage as highly deserving of public encouragement in the prosecution of his arduous undertaking.

Parliamentary Paper No. 370 (May 22 1823).

"Arduous" proved to be no understatement. Government grants totalling £17,000, which is equivalent to over a million euro today and thus a very generous grant, enabled workmen of the highest skill to begin the mechanical construction. However, numerous delays and difficulties occurred, not least because Babbage continually developed new insights to improve the design

after the workmen had started implementation. By 1830 the project had been suspended, Babbage having been accused of dishonesty in misusing Government money (although he was later exonerated by a Royal Society enquiry, which blamed the delay on the Government), having fallen out with his chief engineer, and having aggravated potential allies in government and scientific circles.

However, the main reason for the project's failure was the fact that Babbage was germinating ideas that he could see would render the Difference Engine obsolete. His new Analytical Engine was inspired by the design of Jacquard looms, which were capable of weaving any pattern specified by means of punched holes in pasteboard cards. A manufacturer might use the same cards but different coloured threads in order to vary the colour but not the form of the product. Similarly, the central idea of the Analytical Engine was to separate the operations to be performed from the objects to be operated upon:

> **The Analytical Engine consists of two parts: -**
>
> **1st. The store in which all the variables to be operated upon, as well as all those quantities which have arisen from the result of other operations, are placed.**
>
> **2nd. The mill into which the quantities about to be operated upon are always brought.**
>
> **... There are therefore two sets of cards, the first to direct the nature of the operations to be performed – these are called operation cards: the other to direct the particular variables on which those cards are required to operate – these latter are called variable cards ... The Analytical Engine is therefore a machine of the most general nature ... Every set of cards made for any formula will at any future time recalculate that formula with whatever constants may be required.**
>
> *Charles Babbage (1864), Passages from the Life of a Philosopher, London: Clowes.*

> **It all boils down to what you've got to operate with and how you operate.**
>
> *Mayo Smith, baseball manager (1956).*

The realisation that operations can be represented in a form that machines may interpret was of far greater significance than the engineering of the machine itself.

Babbage first conveyed his ideas about the Analytical Engine to the British Government in December 1834. Babbage himself was in no doubt of the significance of such an invention:

> **Whenever engines of this kind exist in the capitals and universities of the world, it is obvious that all those enquirers who wish to put their theories**

to the test of number, will apply their efforts so to shape the analytical results at which they have arrived, that they shall be susceptible of calculation by machinery in the shortest possible time, and the whole course of their analysis will be directed towards this object.

Charles Babbage (1837), On the Mathematical Powers of the Calculating Engine.

Unfortunately, Babbage had already alienated his potential allies during the Difference Engine project, by, for example, criticizing the Royal Society management, which he considered to be formed of dilettanti rather than active scientists, for its improper conduct of elections. He also criticizing the British Government for under-funding science in general and for funding the wrong projects in particular, which was somewhat tactless given the government's generosity towards his own project.

He then exacerbated his problems with further profound but perhaps unwise speculations. First, Babbage unapologetically described his Analytical Engine in anthropomorphic terms, which some people found offensive:

The analogy between these acts and the operations of the mind almost forced upon me the figurative employment of the same terms. They were found at once convenient and expressive, and I prefer to continue their use rather than to substitute lengthened circumlocutions. For instance, "the engine knows, etc." means that one out of many possible results of its calculations has happened, and that certain changes in its arrangements have taken place, by which it is compelled to carry out the next computation in a certain appointed way.

Charles Babbage (1864), Passages from the Life of a Philosopher, London: Clowes.

In addition, he could not refrain from speculating upon the implications of his engine for subjects other than mathematical analysis. For example, he re-interpreted a paper he had written in 1838 on the nature of miracles in terms of his Analytical Engine:

The workings of machinery run parallel to those of intellect. The Analytical Engine might be so set, that at definite periods, known only to its maker, a certain lever might become moveable during the calculations then making. The consequence of moving it might be to cause the then existing law to be violated for one or more times, after which the original law would resume its reign. Of course the maker of the Calculating Engine might confide this fact to the person using it, who would thus be gifted with the power of prophecy if he foretold the event, or of working a miracle at the proper time, if he withheld his knowledge from those around until the

> **moment of its taking place. Such is the analogy between the construction of machinery to calculate and the occurrence of miracles.**
>
> *Charles Babbage (1864), Passages from the Life of a Philosopher, London: Clowes.*

According to Babbage, his ideas about miracles had been "adopted by many of the most profound thinkers of very different religious opinions."

Babbage thus began three traditions that AI researchers and developers (henceforth, 'AIers', in the absence of a more convenient, established term) continue to the present day. First, he combined grandiose ambition with a failure to deliver – a failure that was quite understandable in the circumstances but provided an easy target for critics. Secondly, he indulged in provocative anthropomorphic descriptions of mechanical devices. Thirdly, he went out of his way to goad those without technical expertise who were concerned about the deeper implications of his work. He surely intended his comments comparing his machinery with the occurrence of miracles to upset Victorian sensibilities – and, naturally, those with a professional interest in such matters reacted:

> **Mr. Babbage, consciously or unconsciously, mixes up mind and matter in a way which is sure to puzzle less philosophic readers. The workings of the intellect and the workings of his machine are always assumed to be conducted on the same principle. Of his engine, we have seen, he speaks habitually as if it were a thinking, reasoning being. It is, to say the least of it, a little startling to hear 'that mechanism had been taught not only to foresee, but also to act upon that foresight.' Note again the marked way in which he always asserts the superiority of mechanism over mind. His automaton is better than a living player at all games of skill. He has, indeed, the grace to confess that the first move must be made by human agency ... But that first move is all that is needed.**
>
> *The Christian Remembrancer (1866).*

These reflections are quite remarkable given that "his automaton" did not even exist. Babbage did speculate on the use of machines to play games such as chess and these speculations seem to have been sufficient for others to attribute to his non-existent machine superiority at all games of skill. The tradition of engaging in philosophical discussions about the implications of machinery before it has been implemented also continues to this day.

Meanwhile, the British Government procrastinated. They did not know what to make of the apparent failure to deliver the Difference Engine and they assuredly did not appreciate the full significance of the Analytical Engine. Even in 1878 its Committee (formed when Babbage's son tried to revive his father's work) felt unable to come off the fence:

> **... having regard to all these considerations, we have come, not without reluctance, to the conclusion that we cannot advise the British Association to take any steps, either by way of recommendation or otherwise, to procure the construction of Mr. Babbage's Analytical Engine and the printing tables by its means.**
>
> *C.W. Merrifield (1878), Report of the Committee appointed to consider the advisability of constructing Mr. Babbage's Analytical Machine.*

Increasingly embittered, Babbage continued to squander his personal fortune on trying to construct the Analytical Engine:

> **A short time after the arrival of Count Strzelecki in England, I had the pleasure of meeting him at the table of a common friend. Many enquiries were made relative to his residence in China. Much interest was expressed by several of the party to learn on what subject the Chinese were most anxious to have information. Count Strzelecki told them that the subject of most frequent enquiry was Babbage's Calculating Machine. On being asked further as to the nature of the enquiries, he said they were most anxious to know whether it would go into the pocket ... I told the Count that he might safely assure his friends in the Celestial Empire that it was in every sense of the word an *out-of-pocket* machine.**
>
> *Charles Babbage (1864), Passages from the Life of a Philosopher, London: Clowes.*

His efforts at building the Analytical Engine provoked derision among his contemporaries. He died in 1871, unlamented and misunderstood.

During his lifetime only Ada Lovelace had fully appreciated his genius and indeed it is mainly through her interpretations of his ideas that we may do so today, in particular through her notes written in 1843 accompanying her translation of Louis Menebrae's paper on Babbage's Analytical Engine. These 'notes' are rather more substantial than the word suggests, being three times longer than the paper itself. She had died in 1852, aged 36, of cervical cancer and driven to despair by gambling debts. Her correspondence with Babbage, parts of which he destroyed, suggests that they hoped to apply the calculating machines to devise a system for winning bets on horse races.

3. The first computers: "electronic brains"

The sad demise of Ada Lovelace indicates that an addiction to gambling was an occupational hazard for unemployed, rich ladies of the time. Occasionally, the outcome was more positive. The wife (Mary Shelley) of the friend (the poet, Percy Bysshe Shelley) of the father (Lord Byron) of Ada Lovelace wrote *Frankenstein, or The Modern Prometheus* in 1818 as a result of a

wager. The eponymous anti-hero of *Frankenstein* created a synthetic man by galvanising human fragments gathered from graveyards and dissecting rooms. The synthetic man, not dignified with a name by Mary Shelley, was shunned by everyone and eventually took retribution on its or his creator.

A similar outcome awaited the creator of a chess-playing machine in Ambrose Bierce's *Moxon's Master* written in 1893: when he defeats the machine it rises up in anger and strangles him. This is a common fable, warning us of the dangers of creating uncontrollable machines that will inevitably lead to malevolence, conflict, and the destruction of mankind, or man, at least.

Other fiction writers also took the opportunity to explore ethical questions concerned with the comparisons between man and machine provoked by the calculating devices and other machinery. For example, Frank Baum's *Wizard of Oz* stories included a mechanical man called Tiktok who could be wound up to think, speak and walk, and as a result:

> **... was so trustworthy, reliable and true; he was sure to do exactly what he was wound up to do, at all times and in all circumstances. Perhaps it is better to be a machine that does its duty than a flesh-and-blood person who will not, for a dead truth is better than a live falsehood.**
>
> *L. Frank Baum (1909), The Road to Oz, New York: Dover Publications.*

The biologist Thomas Henry Huxley had already come to a similar conclusion, for he had offered to concede his free will if he could be made a clockwork man guaranteed to do what is right:

> **If some great Power would agree to make me always think what is true and do what is right, on condition of being turned into a sort of clock and wound up every morning before I got out of bed, I should instantly close with the offer. The only freedom I care about is the freedom to do right; the freedom to do wrong I am ready to part with on the cheapest terms to any one who will take it from me.**
>
> *Thomas Henry Huxley (1893), Methods and Results, London: Macmillan.*

Huxley was known as 'Darwin's bulldog' because he did more than anyone else to argue the case for Darwin's theory of evolution, and possibly this coloured his view of the importance of free will.

As far as real computing machines are concerned, our story pauses with Babbage's death. Babbage (with Ada Lovelace) had demonstrated the link between material, mechanical operations and abstract, mental conceptions and had thereby stimulated the first debates on what was to become known much later as AI. The earlier calculating devices had already shown that metal could do some things which had previously been restricted to the human brain but the Analytical Engine went beyond this by providing a

general means of manipulating arbitrary algebraic and numeric symbols. Although the mechanical technology of the time was inadequate to construct the Analytical Engine, Babbage had identified a number of conceptual innovations: he had established the structure of calculating machines (with arithmetic units, memories, control units, and input-output devices); he had distinguished between operators and operands (that is, the things operated on); he had focussed on digital, rather than analogue, machines. In short, Babbage had invented the concept of a universal computing machine and discovered the idea of computer programming.

However, Babbage missed two insights that, in retrospect, may seem surprising. First, he did not take the strong hint provided by the fact that he had been able to represent both operators and operands in identical form, as holes in punched cards. This implied that they were fundamentally the same thing – which has profound implications, to be discussed shortly. Secondly, he did not clearly distinguish between the concept of the Analytical Engine and the means by which it might be implemented. There is no evidence that Babbage ever saw his engines as anything other than purely mechanical ones, despite his difficulty in constructing them. In particular, he did not think of using electricity, although the early nineteenth century was a period of rapid development in the understanding of electrical circuits, with Michael Faraday building the first electric motors in 1821.

The distrust of electricity continued for some time:

> **When a bet is made the ticket seller depresses a key bearing the number of the particular horse, and thus, as in other makes of totalisator, closes a circuit. As soon as current passes, a steel ball of 5/8 inch diameter is released from a receptacle contained in the main machinery ... it is claimed … that this practice of using steel balls as the main link between the selling and integrating sections of the machines gives greater assurance of positive action than a purely electrical cycle of impulses.**
>
> *D.H.N. Caley (1929), Electricity and the "Tote".*

Electricity had first been used for a major 'information processing' task in the 1890 US census, thanks to the invention by Herman Hollerith of an electromagnetic device to read punched cards. In 1896 Hollerith set up the Tabulating Machine Company, which became IBM in due course.

After Babbage, the technology of computer design stalled for almost a hundred years until the Second World War provided an impetus to, for example, decode coded messages and produce firing tables. Of course, the design of calculating machines continued to be refined, by, for example, using binary rather than decimal numbers, as had been anticipated by Leibniz. George Stibitz of Bell Laboratories first implemented this in 1937.

However, the concept of a universal computer was not revived until the war years, which is a shame for the history of computing because it means that much of the early development occurred in secrecy in laboratories in the US, the UK, Germany and France. Rumours of this classified research led, as usual, to some exaggeration, with one machine being described as:

> **… able to solve problems which man had no hope of solving, in physics, electronics, atomic structure and, who knows, perhaps even to solve the problem of the origin of mankind.**
>
> *L.J. Comrie (1944), American Weekly, October 14.*

This comment referred to what is now known as the Harvard-IBM machine, probably the first universal calculator, if not quite a computer (because it did not store a program in its memory, now taken to be a defining characteristic of a computer, as discussed shortly).

The blame for such sensationalism cannot be put entirely on journalists and publishers, for they have a professional obligation to stimulate interest in their topic. This tends to lead to an emphasis on the extreme views, that computers are either miraculously brainy or plain dumb, as Maurice Wilkes, the director of the first Computing Laboratory at Cambridge, England, discussed:

> **Two contrary attitudes are common. In the first place there is a widespread, although mostly unconscious, desire to believe that a machine can be something more than a machine, and it is to this unconscious urge that the newspaper articles and headlines about mechanical brains appeal. On the other hand, many people passionately deny that machines can ever think. They often hold this view so strongly that they are led to attack designers of high-speed automatic computing machines, quite unjustly, for making claims, which they do not in fact make, that their machines have human attributes.**
>
> *Maurice Wilkes (1953), Can a machine think?, Discovery.*

So, even by 1953, it was "common" to find attitudes about electronic brains that can never think.

However, it is unfortunately not the case that it was unjust to accuse designers of the new machines of describing them in human-like terms. For example, John von Neumann, now regarded as the father of the modern computer (so much so that the standard design is often described today as a 'von Neumann machine'), drew analogies with the human nervous system, as it was then understood:

> **The three specific parts ... correspond to the associative neurons in the human nervous system. It remains to discuss the equivalents of the sensory**

or afferent and the motor or efferent neurons. These are the input and output organs of the device.

John von Neumann (1945), in Herman Goldstine (1972), The Computer from Pascal to Von Neumann, Princeton, N.J.: Princeton University Press.

Given that by this time there were scores of scientists working on computing machinery, the term 'von Neumann machine' may be seen as an acknowledgement that John von Neumann was the one true genius amongst them, for as Jacob Bronowski remarked in *The Ascent of Man* (1973) a genius is a man who has *two* great ideas. Von Neumann had already established his reputation as a mathematician and, in particular, with his 1944 book with Oskar Morgenstern on the *Theory of Games and Economic Behavior*, where by 'game' he meant real-life decision-making not artificial games like chess. His fascination with computers derived partly from a concern to show that human decision-making was different from the precise calculations of mathematics and engineering, as he elaborated in *The Computer and the Brain* (1956). Still, when the leading specialists in the field say that the components of a computer correspond to neurons the technologically unenlightened cannot be blamed for referring to 'electronic brains', as they began to do in the 1940s and 1950s.

Apart from potentially misleading the general public, such analogies were liable to be misinterpreted on invalid technological grounds:

A computer with as many vacuum tubes as a man has neurons in his head would require the Pentagon to house it, Niagara's power to run it, and Niagara's waters to cool it.

Warren McCulloch, quoted in Robert Lindner (1956), Must You Conform?, New York: Holt, Rinehart & Winston.

This is not to imply that Warren McCulloch was technologically unenlightened. He was a neurophysiologist who had written, with Walter Pitts, a seminal paper on *A Logical Calculus of the Ideas Immanent in Nervous Activity* in 1943. Their proposition was that philosophical problems could be solved only in terms of the physiology of the nervous system. To this end they developed a mathematical model of neural nets, that is, of neurons and the activities between them, and encouraged a view of the brain as a computing machine. While its model of the neuron was soon shown to be overly simple, the 1943 paper did initiate the field of neural net research, which continues to this day. It certainly influenced von Neumann, who adopted it to teach the theory of computing machines.

4. Cybernetics: "practically infinite modes of existence"

Physiological ideas about how feedback operates in the human body were also instrumental in the development of the subject of cybernetics, a word coined by Norbert Wiener in 1948 from the Greek kubernetes, meaning steersman (at least, it is attributed to Wiener although Ampère had referred to the science of government as *la cybernétique* in 1843). Wiener had co-authored perhaps the first paper on the subject, *Behaviour, Purpose and Teleology*, published in 1943. Having been raised by his father to be a genius, Wiener gained his PhD from Harvard University at the age of 18 and played his allotted role with enthusiasm, developing a penchant for engaging in feuds, even with friends, over scientific disagreements. In 1945 he and von Neumann established the Teleological Society to discuss the topics that were coalescing to form cybernetics. Apart from the ferment of new ideas, the meetings must have been fascinating for the contrast in personalities. Von Neumann was urbane and sociable. He went on to advise political leaders, using his technological expertise to argue, for example, for a preventive attack against the USSR. In contrast, in 1946 Weiner vowed to withdraw from any research that might be misused by "irresponsible militarists".

In the same year (1948) that Wiener published his book on cybernetics, Claude Shannon published *A Mathematical Theory of Communication*, outlining what is now known as information theory, which is concerned with the transmission of data through unreliable channels and which forms the basis of today's telecommunications:

> **Great scientific theories, like great symphonies and great novels, are among man's proudest – and rarest – creations … Within the last five years a new theory has appeared that seems to bear some of the hallmarks of greatness … It may be no exaggeration to say that man's progress in peace, and security in war, depends more on fruitful applications of information theory than on physical demonstrations, either in bombs or power plans, that Einstein's famous equation works.**
>
> *Claude Shannon (1953), The Information Theory.*

Shannon had already become famous for his 1937 thesis that provided a formal way of describing electrical circuits and that was immediately applied to the development of telephone systems. He diffidently considered himself lucky that he was the only one at the time familiar with the academic fields of both mathematics and electromagnetism, and he had a reputation as a man lacking in ego (as compared to Wiener perhaps) – a reputation that does not quite square with the above quotation.

Clearly, there were theoretical and practical problems in the air in the 1930s and 1940s that were attracting the brightest intellects, such as von Neumann, Wiener, McCulloch, Pitts, Shannon and others. The common thread to all their endeavours is the gradual recognition that 'information' may be added to matter and energy as one of the fundamental properties in terms of which the universe may be described. With an appropriate definition, the concept of information is relevant to the description of electrical circuits, communications between computers and people, arrangements of atoms in biological molecules, and memory encodings in brain cells.

Cybernetics itself now appears to have been a motley collection of theories of linear servomechanisms, information and nerve networks. In fact, the digital computers that were eventually used to simulate intellectual processes were not derived from cybernetic theories, but it is interesting to see the eagerness with which the modest technology used in early cybernetic machines was imbued with human-like properties. For example, four small bar magnets swinging over a battery were described in these terms:

> **Ross Ashby who built the machine thinks it is the closest thing to a synthetic human brain so far designed by man ... Dr. Ashby does not consider his first homeostat particularly intelligent, but he feels sure that a really bright model can be built on the same principle.**
>
> *Time Magazine (1950), The Thinking Machine.*

Similarly, the neurophysiologist Grey Walter was inspired by his 'tortoises' Elsie and Elmer, which each had a couple of photoelectric cells for vision, motors to drive the wheels, a few capacitors to provide memory, and some nerve-like connections between all these, to conclude:

> **The fact that only a few richly connected elements can provide practically infinite modes of existence suggests that there is no logical or experimental necessity to invoke more than a number to account for our subjective conviction of freedom of will and our objective awareness of personality in our fellow man.**
>
> *W. Grey Walter (1951), A machine that learns, Scientific American, 184, 8, 60-63.*

Of course, cybernetic machines do not have to be built with such primitive technology, no more than computing machines have to be built with vacuum tubes.

As we now know, computer technology has developed at a bewildering pace:

> **It takes no prophet, however, to note the vast potentialities in improved reliability, in decreased power, space, heat, and weight, and very likely in**

> **increased capacity and decreased costs promised by the various solid-state devices, the transistors, the magnetic cores, and the ferro-dielectrics.**
>
> *C.W. Adams (1954), Small computers in a large world, Proceedings of the Eastern Joint Computer Conference.*
>
> **The "complete computer on a chip" appears to be realizable within one or two years ... before 1975 what we will call the "microcomputer" will require only one chip and cost less than $100 in small quantities ... We predict a microcomputer [in 1997] will be able to execute at the rate of 10 mips [million instructions per second]. To sum up, we have the equivalent of a 7090 (without any peripherals) on one chip at a cost of about $1.00 ... in 1960, IBM was able to produce the 7090 for approximately $3 million.**
>
> *Caxton C. Foster (1972), A view of computer architecture, Communications of the ACM, 15, 557-565.*

It is possible to be bedazzled by a comparison between the physical properties of early and modern computers. If they were discussing electronic brains in the 1950s and our computers today are orders of magnitude better in all technical respects, then surely they must be much brainier. However, as Dijkstra warned us:

> **Modern-day computers are amazing pieces of equipment, but most amazing of all are the uncertain grounds on account of which we attach any validity to their output.**
>
> *Edsger Dijkstra (1972), Notes on Structured Programming, in O.J. Dahl, E.W. Dijkstra and C.A.R. Hoare (eds.), Structured Programming, London: Academic Press, 1-82.*

the important questions concern what computers do and cannot do, not what their physical properties are.

5. The Turing machine: "an effective procedure"

Computer technology is indeed amazing but in theoretical terms is irrelevant to the question of artificial intelligence. Even before the first computer had been built, the question of 'what is computable?' had been considered and it had been concluded that the answer did not depend on the computer machinery itself. Given sufficient time, any computer will compute exactly the same set of functions as any other computer. The American logician Alonzo Church and the British mathematician Alan Turing had independently attempted the definition of this set in the 1930s. Since their definitions, and those of others subsequently, turn out to be equivalent it is now generally accepted that the intuitive notion of computability is captured

by saying that a function is computable if and only if it can be determined (in principle) by a particularly simple machine devised by Turing in 1936.

Such a 'Turing machine' consists of

(1) an indefinitely extendible tape marked off in squares (like a perforated toilet roll but with the handy property of being as long as is needed);

(2) a tape unit, which can look at only one square at a time, write a symbol chosen from a finite set of symbols on a square, and move the tape one square in either direction;

(3) a control unit, which can assume only one of a finite number of states.

The machine computes via a sequence of discrete steps, each of which is completely determined by the symbol currently being read and the state the control unit is in. A step involves three activities: writing a symbol, moving the tape one square, and entering a new state or halting. Stating these three components of a step for each symbol-state combination defines a computation for a Turing machine. A computation expressed in these terms is an 'effective procedure', that is, one which is so clearly and precisely defined that it may be carried out without any interpreting initiative or intelligence:

> **An effective procedure is a set of rules which tells us, from moment to moment, precisely how to behave.**
>
> *Marvin Minsky (1967), Computation: Finite and Infinite Machines, Englewood Cliffs, N.J.: Prentice-Hall.*

Thus the Turing machine is intended to idealise the fundamental properties that any computer must possess: a finite, deterministic set of instructions and a large data store.

Turing machines are not physically constructed, not because they are too complicated (like Babbage's machines) but because they are too simple. They are idealised computers. Any real computer can (in principle) be simulated by a Turing machine, and vice versa:

> **This special property of digital computers, that they can mimic any discrete state machine, is described by saying that they are universal machines. The existence of machines with this property has the important consequence that, considerations of speed apart, it is unnecessary to design various new machines to do various computing processes. They can all be done with one digital computer, suitably programmed for each case.**
>
> *Alan Turing (1950), Computing machinery and intelligence, Mind, 59, 236, 433-460.*

Ultimately, therefore, arguments about whether or not a particular function is computable (or programmable) can be reduced to questions about Turing machines. It seems therefore that the question of whether intelligence is

computable is reducible to a theoretical analysis of Turing machines, which may be seen as a relief, since we don't need to worry about complicated real computers, or as a disappointment, since Turing machines seem so uninspiring. However, the Church-Turing hypothesis – that computation corresponds exactly to what a Turing machine can do – is just that, a hypothesis. It is neither a theorem amenable to mathematical proof nor a scientific law open to experimental study. It is a matter of whether or not we agree that the intuitive notion of computation has been captured.

While most computational theorists are prepared to agree, some questions can be raised, apart from the obvious one of whether a theoretical analysis of an 'intelligent' Turing machine, which would be inconceivably complex, is actually possible. For example,

- The Turing machine seems rather enclosed, with its computation steps and data all available at the outset. What about if the data only becomes available during the computation and hence may be unpredicted by the computation? For humans, intelligent behaviour seems to involve the reaction and adaptation to unanticipated events.
- What if the computation steps themselves are not specified fully in advance but are somehow created during the computation?
- The Turing machine seems to have a 'discrete' view of the world, with data being made available piece by piece. Is it possible to have computers that receive data in a more continuous way? We do not appear to receive sight and sound data discretely.
- Although the physical structure of a Turing machine is irrelevant, there seems to be an assumption that conventional physics applies to it. When computer components become very fast and very small, so that quantum effects may be important, does that make a difference? We know that at the quantum mechanical level predictability disappears.
- The Turing machine is assumed to work in splendid isolation, whereas we know that in practice real computers are networked. Does that matter? Are the properties of a crowd of people fully determined by the properties of its individual members?
- It seems to be assumed that the computation of a Turing machine is the determination of an output result from input data, for example, the calculation of the square root of a given number. But many computations do not have results; they just run. For example, they monitor a power station or a heart pacemaker. Does that make a difference?
- What about computers that function symbiotically with humans? At the moment, we tend to view this combination as a human with a computational aid, such as a speech-generator or prosthetic arm. Is it possible to look at

the situation from the point of view of the computer? If a computer were able to see via a human eye would that make a difference to its theoretical properties?

These are deep questions that we cannot begin to answer now or indeed do more than scratch the surface of later: they are raised now to make it clear that matters are not necessarily as straightforward as Turing machines may suggest. Some AIers go so far as to consider that:

> **It is … not just wrong but essentially meaningless to speculate on the ability of Turing machines to be able to perform human-like intelligence tasks.**
>
> *Sunny Bains (2003), Intelligence as physical computation, AISB Journal, 1, 225-240.*

This opinion derives mainly from the view that Turing machines cannot respond to an external event in order to display so-called natural behavioural intelligence.

Certainly, the analysis of intelligence via Turing machines is not an avenue that has been followed even by pure mathematicians since Turing machine programs for any function of significance are extraordinarily detailed and inscrutable. Real programming is, of course, carried out at a much higher level of abstraction. For now, the main point is that the quest for artificial intelligence is largely independent of developments in computer hardware: it is the nature of computer software that is the issue.

6. The stored program concept: "stored in the memory"

We left the concept of a 'program' with Babbage and his two sets of cards, one for operations and one for operands. The idea that these two sets can be treated equally eventually occurred:

> **Babbage's Jacquard-system and mine differ considerably; for, while Babbage designed two sets of cards – one set to govern the operations, and the other set to select the numbers to be operated on – I use one sheet or roll of perforated paper … to perform both these functions in the order and manner necessary to solve the formula to which the particular paper is assigned. To such a paper I apply the term formula-paper ... of course, a single formula-paper can be used for an indefinite number of calculations, provided that they are all of one type or kind.**
>
> *P.E. Ludgate (1907), On a Proposed Analytic Machine.*

This insight leads directly to the 'stored program concept', which is the key design feature of modern computers. With Babbage's Analytical Engine the values to be operated on were read in and stored within the machine and the operations were read in one by one when required. This limited the speed of

the machine and also made it difficult to repeat sequences of operations – either the cards were duplicated or some means provided to re-read previous cards. However, if the operations were read in and stored in the machine, just as the operands were, then they would be instantly available to the machine to be carried out whenever required – provided, of course, that it could be arranged for the machine to find them:

> **It is evident that the machine must be capable of storing in some manner not only the digital information needed in a given computation ... but also the instructions which govern the actual routine to be performed ... conceptually, we have discussed above two different forms of memory: the storage of numbers and the storage of orders. If, however, the orders to the machine are reduced to a numerical code and if the machine can in some fashion distinguish a number from an order, the memory organ can be used to store both numbers and orders.**
>
> *Arthur W. Burks, Herman H. Goldstine and John von Neumann (1946), Preliminary Discussion of the Logical Design of an Electronic Computing Instrument, Princeton, Institute for Advanced Study, in John von Neumann (1963), Collected Works, Oxford: Pergamon Press.*

With electronic computers, storing the program in this way enormously increases the speed of execution. Also, since the operations are stored in exactly the same way as the operands, there is no reason why the machine could not change the operations just as it is able to change the operands. It is possible therefore to write self-modifying programs, as we will see.

The idea of a stored program is the defining one for a computer:

> **The name *computer* is used to mean a universal calculator in which the program is stored in the memory.**
>
> *René Moreau (1981), Ainsi naquit l'informatique, Paris: Bordas.*

It would, therefore, be desirable, in the interests of historical accuracy, to identify to whom it should be attributed. Clearly, Burks, Goldstine and von Neumann had the stored program idea by 1946. However, the earlier *First Draft Report on the EDVAC*, a machine explicitly designed to minimise the use of resources, also discussed the use of one memory for both data and program. This report was revised and edited by von Neumann, whose name, as first editor, became most associated with the report and hence with the stored program concept. Today, though, it is generally conceded that the credit should go to the authors of the first draft, John Mauchly and J. Presper Eckert.

But not everyone concedes the invention of the computer to Mauchly and Eckert. In 1973 John Atanasoff of Iowa University won a legal case to establish that he had invented the computer, by means of his ABC machine

built some time in the period 1939-42. However, ABC was a special-purpose machine. It could not be programmed to perform general functions and as it had no programs they could not, of course, be stored. A case can also be made for Konrad Zuse in Germany who in 1936 filed a patent that at least came close to the idea:

> **The invention serves the purpose of automatic execution by a computer of frequently recurring computations, of arbitrary length and construction, consisting of an assembly of elementary arithmetic operations. The prerequisite for each type of calculation that is to be done is to prepare a computation plan in which the operations are listed serially indicating their type ... once the computation plan has been set out for a specific problem, it is valid for all other input values as well ... the computation plan is represented in a form suitable for the control of the individual devices, e.g. on a punched tape.**
>
> *Konrad Zuse (April 11 1936), Patent Application Z23 139 IX/42m: Method for Automatic Execution of Calculations with the Aid of Computers.*

Zuse is now credited with having built the first program-controlled computer, the Z-3, in 1941. The first electronic, general-purpose computer, ENIAC, was built in the United States by Mauchly and Eckert in 1946 and the first stored program computer was either their BINAC or, more likely, Maurice Wilkes's Cambridge computer, EDSAC, both completed in 1949.

7. Programs: "nothing must be left unstated"

A 'computation plan', henceforth 'program', therefore consists of a serial list of operations to be performed. The machine to which this list is presented must be designed to be able to carry out those operations:

> **He'll sit there all day saying "Do this!" Do that!" and nothing will happen.**
>
> *Harry S. Truman (1952), referring to the 'government machine', in contrast to the 'Army machine', with which his successor, Dwight Eisenhower, was more familiar.*

It was soon found out that even if anything did happen it was often the wrong thing because it was difficult to specify the operations correctly. Within a short period, this little difficulty had evolved into "the greatest intellectual challenge that mankind has faced":

> **Computer programming as a practical human activity is some 25 years old, a short time for intellectual development. Yet computer programming has already posed the greatest intellectual challenge that mankind has faced in pure logic and complexity. Never before has man had the services of such logical servants, so remarkable in power, yet so devoid of common sense**

> **that instructions given to them must be perfect, and must cover every contingency, for they are carried out faster than the mind can follow.**
>
> *Harlan D. Mills (1975), The new math of computer programming, Communications of the ACM, 18, 43-48.*

According to this view, computer programs needed to be correct and complete because computers lacked the 'common sense' to work out what we really intended.

So, communicating with a computer was regarded as the opposite of communicating with fellow humans, who are, of course, well endowed with common sense:

> **One could almost say that the rules for writing mathematics for human consumption are the opposite of the rules for writing mathematics for machine consumption.... For the machine, nothing must be left unstated, nothing must be left "to the imagination." For the human reader, nothing should be included which is so obvious, so "mechanical" that it would distract from the ideas being communicated.**
>
> *Philip J. Davis and Reuben Hersh (1981), The Mathematical Experience, Boston: Birkhäuser.*

But, while nothing must be left unstated, it does not follow that it has to be stated directly by the human programmer.

Certain elementary operations, such as the addition of two integers, will be built-in, that is, electronic circuitry will be provided to carry them out. Because the circuitry is relatively expensive, the number of built-in operations will be small, typically of the order of a hundred or so, and consequently the operations themselves very simple and general. All programs have eventually to be expressed in terms of the built-in operations because, by definition, that is all the computer can carry out.

Programming in terms of built-in operations is, however, very tedious and error-prone. It is much more convenient to program in terms of more complex operations which are converted by the computer itself into the built-in operations. Which operations are built-in is a designer's decision: multiplication, for example, could be built-in or converted into a series of built-in additions:

> **A decision must be made as to which operations shall be built in and which are to be coded into the instructions ... many operations which are thus excluded from the built-in set are still of sufficiently frequent occurrence to make undesirable the repetition of their coding in detail. For these, magnetic tapes containing the series of orders required for the operation can be prepared once and for all and be made available for use when called for in a particular problem. In order that such subroutines, as they can**

> **well be called, be truly general, the machine must be endowed with the ability to modify instructions, such as placing specific quantities into general subroutines.**
>
> *John Mauchly (1947), Preparation of Problems for EDVAC-type Machines, Proceedings of a Symposium on Large-Scale Digital Calculating Machines, in Annals of the Harvard Computation Laboratory (1948), 16, 203-207, Cambridge, Mass.: Harvard University Press.*

This is believed to be the first discussion of the concept of a 'subroutine', to be discussed further below.

If all the operations which are executed have been explicitly or implicitly specified by a programmer then it would seem to be the case that all computers can do only what they have been programmed to do and hence that any intelligence apparent in the computer's performance should be attributed to the programmer – a kind of inverse of the slogan "Garbage in, garbage out", that is, "Intelligence out, intelligence in". Consequently, no computer can be more intelligent than its programmer, as the very last paragraph of one of the first general reviews of computers reassured us:

> **The limitations of man as a programmer will always, in the end, set a limit to the intelligence that can be simulated by a machine. A computer's 'artificial intelligence' is prescribed by man, and a higher intelligence is demanded for its prescription than for the execution. Man, as the originator, will always be on top.**
>
> *S.H. Hollingdale and G.C. Tootill (1965), Electronic Computers, Harmondsworth: Penguin.*

This argument, which will see to be not as straightforward as it seems, predates computers:

> **[Descartes had been] led astray by the idea that the automaton must do its own reasoning, whereas it is the builder of the automaton who does the reasoning for it.**
>
> *Leonardo Torres y Quevedo (1914), Essais sur l'automation, Revue de l'Academie Royale de Madrid.*

Torres y Quevedo, a Spanish engineer, is not well-known today but he is an important figure in the history of AI, for apart from these speculations on its limitations he was the first person to show (by actually building some of the components) that Babbage's Analytical Engine could be built using electromagnetic technology, he was the first person to build a game-playing machine (which played chess end-games), and is also said to be the first person to use the term 'automation' for the new science and to give a definition:

> **The main aim of automation is that the automata be capable of discrimination; that at any moment they be able to take into account the impressions they are receiving, or have received up to that time, in performing the operations then required. They must imitate living beings in regulating their actions according to their impressions and in adapting their conduct to the circumstances of the time.**
>
> *Leonardo Torres y Quevedo (1914), Essais sur l'automation, Revue de l'Academie Royale de Madrid.*

He therefore gave more emphasis to adaptive, interactive nature of automata than, for example, Turing machines appear to do.

The idea of a subroutine, discussed by Mauchly (above), is central to modern programming methodologies. A subroutine is a self-contained package of operations that can be thought of as a single operation itself. Some of the operations within a subroutine may themselves be subroutines and consequently we may build a hierarchy of ever more detailed subroutines.

Programming then involves taking a complex problem and progressively breaking it down into simpler sub-problems (subroutines) until built-in operations are reached. To carry out or execute a subroutine, it is 'called' from a point in the calling routine; when the subroutine finishes it returns, with a result if there is one, to that point in the calling routine, which then continues:

> **Life makes no absolute statement. It is all call and answer.**
>
> *D.H. Lawrence (1923), Kangaroo, London: Secker.*

As Mauchly observed, a subroutine that always carried out exactly the same set of operations is not particularly useful: when cooking a meal, account should be taken of the number of diners. When a subroutine is called, information is passed to it to enable it to vary its operation, for example, by saying that 'number of diners = 8'. The definition of the subroutine has to be in terms of 'variables' (such as 'number of diners') that only take values when the subroutine is called.

The generality and usefulness of programs is considerably increased by the judicious use of such variables:

> **Once a person has understood the way in which variables are used in programming he has understood the quintessence of programming.**
>
> *Edsger Dijkstra (1972), Notes on Structured Programming, in O.J. Dahl, E.W. Dijkstra and C.A.R. Hoare (eds.), Structured Programming, London: Academic Press, 1-82.*

Without variables, a program is an unvarying sequence of steps. With variables, the steps of a program may be contingent on the values that the

variables take when the program is executed. In a typical programming notation we would write:

```
if [condition]
   then [operations]
   else [some other operations]
```

with the 'else' part being optional. For example:

```
if you cannot stand the heat
   then get out of the kitchen
```

might be an instruction for a domestic robot.

Babbage recognised the need for such conditional instructions in the design of his Analytical Engine, as he realised that it would be necessary for the machine to move to different parts of the program, depending on the result of some test. Simple as such instructions may seem they fundamentally change the properties of computers by making them less predictable. The question "What does it do?" has to be answered with an equivocal "It depends":

"I only said 'if'!" poor Alice pleaded in a piteous tone.

Lewis Carroll (1871), Through the Looking Glass.

Indeed, if the conditional instructions are numerous and complex, it is easy to imagine that the programmer herself may have difficulty in predicting how the computer will perform in any particular situation.

One use of conditional instructions is to enable a set of operations to be repeated, until some condition becomes satisfied: for example, "simmer the escallops in the milk until tender". Typical forms of expression in programming languages are

```
repeat [operations] until [condition]
while [condition] do [operations]
```

For example, an instruction to a robotic mouse might be:

```
while the cat is away do play
```

Most computations are built with such iterative instructions, which again can become quite complex:

O! Thou hast damnable iteration and art,
Indeed, able to corrupt a saint.

William Shakespeare, Henry IV, 1, 1.

Conditional and iterative instructions are adequate for the writing of any computer program, but it will be convenient to organise them within subroutines.

Like history, a subroutine may repeat itself. If the set of operations to be repeated is encapsulated in a subroutine, then it is possible for the subroutine to call itself. Such subroutines are said to be 'recursive'. They

are particularly useful in AI programming, as it often happens that problems are broken down into sub-problems that are of the same form as the original problem, as we will see.

Imagine, for example, that someone has jumbled up your Rubik cube and you want to return it to its original, 'solved' state. (For younger readers, a Rubik cube was a puzzle that was a craze in the 1970s. Each face of the cube consisted of three rows and three columns, each of which could be independently rotated about the cube's axis. In a solved state, all nine squares on each of the six faces had the same colour. In its typical state, that is, lost at the back of a cupboard, it had a random arrangement of the six colours over the faces.) You might be reluctant to start shuffling it about as you suspect that the cube is only a little jumbled and you might only make it worse. So you decide to write a program which, given a description of the jumbled state, will tell you the shortest sequence of twists which will sort it out. A recursive subroutine to solve a Rubik cube problem by randomly twisting a layer might be:

```
to solve Rubik cube problem X:
   if X is actually the solution then stop
   else  twist any layer to get a new problem Y and
         solve problem Y.
```

As it stands, this subroutine is far from guaranteed to find the shortest solution or indeed any solution at all. Moreover, it is little more than the re-phrasing of a simple loop. The humorist Gerard Hoffnung used to tell a long anecdote in which (in summary) he went to a shop and bought a Swiss army knife but on reaching home found he couldn't open it so he returned to the shop to buy a Swiss army knife to open it but on reaching home ... It was a very long anecdote.

To be of any use, a recursive subroutine has to be concerned with a simpler version of the original problem. Imagine, for example, that you want to write a subroutine that, given an integer stored in binary (e.g. 101100101011), should print the decimal digits (2849, in this order), using only a subroutine to output a single digit. A recursive subroutine to achieve this is:

```
to print-the-integer n:
   if n<10 then output the digit n
   else
      print-the-integer (n/10 ignoring the remainder)
      output the digit which is the remainder of n/10.
```

This subroutine acknowledges that it is easier to get hold of the 9 (which is the remainder when the number is divided by 10) than it is the 2, because we

don't, in general, know how big the number is, but, of course, we can't output the 9 until we have dealt with the 284. In this case, the subroutine calls itself three times before 'unwinding' to print the digits 2, 8, 4 and 9.

This illustrates the standard form of a recursive subroutine: there has to be at least one terminating condition and at least one recursive call with a simpler version of the problem. Another example is the following subroutine to solve the Tower of Hanoi problem, which is to move n disks (initially in a pile from the smallest, numbered 1, to the biggest, numbered n, on peg A) from A to peg B, using peg C as a spare, without ever putting a disk on a smaller one:

```
to solve Tower of Hanoi problem A, B, C, n:
   if n=0 then stop
   else
      solve Tower of Hanoi problem A, C, B, n-1
      move disk n from A to B
      solve Tower of Hanoi problem C, B, A, n-1.
```

According to legend, when the monks in a monastery near Hanoi solve this problem the world will come to an end.

Computer programs consist essentially of sets of conditional and iterative instructions organised into subroutines, some of them recursive. Most programs consist of thousands of lines of text defining these instructions:

> **A good program can be read like a book, from the beginning to the end without turning pages back and forth looking elsewhere for an explanation of what is going on.**
>
> *Per Brinch Hansen (1977), The Architecture of Concurrent Programs, Englewood Cliffs, N.J.: Prentice-Hall.*

If it were the case that programs could be read and understood like novels then we might easily conclude that the meaning of the program (or novel) and therefore any intelligence that we might attribute to it has been determined solely by the programmer (or author) and cannot be attributed to the computer (or book).

However, matters are not so clear-cut. According to the above description of subroutines, the programmer says when a subroutine should be called by specifying it by name in the appropriate place: so the programmer has to work it all out in advance. In most modern programming languages, however, subroutines are not called by name but by the program broadcasting a 'message' while it is running, indicating what it wants to achieve, and by all the available subroutines (called 'objects', in this context) considering whether they may contribute to what is to be achieved. The difference is

roughly that between a restaurant where you say "François, more chardonnay, please" and one where you hold up a board saying "I'm thirsty" and hope that some waiter will respond appropriately. In the latter case it is much harder, although it can be done in principle, to predict how the program will behave. The point is that the programmer is no longer thinking in detail of what the program will do when it runs.

The staple diet of ordinary programming is given different flavours in the numerous notations that have been devised to express programs. The design of these programming languages is a central concern of 'computer science', now earnestly studied in all universities. It surely can only be employment prospects which attracts growing numbers to computer science, for its core activity of programming seems particularly dreary, involving as it does the tedious specification of all the minute details needed for a program to function properly. A sideways perspective, perhaps corresponding to that of the general population, compares a computer scientist to someone with a diseased right hemisphere:

> **With individuals who have disease of the right hemisphere, the abilities to express oneself in language and to understand ... others are deceptively normal. These patients are strangely cut off from all but the verbal messages of others ... they are reminiscent of language machines ... appreciative of neither the subtle nuances or non-linguistic contexts in which the message was issued ... Here the patient (with right hemisphere damage) exemplifies the behaviour ... associated with the brilliant young computer scientist. This highly rational individual is ever alert to an inconsistency in what is being said, always seeking to formulate ideas in the most airtight way; but in neither case does he display any humor about his own situation, nor ... the many subtle intuitive interpersonal facets which form so central a part of human intercourse.**
>
> *Howard Gardner (1974), The Shattered Mind: The Patient after Brain Damage, New York: Knopf.*

8. Programming: "an aesthetic experience"

The prevalent utilitarian view of computer science and programming languages as just the means of building practically useful machines (and thereby gaining lucrative employment) does not capture the unique nature of computers – that they address the question of whether and how we might express our thought processes in a precise form:

'Computer science' is not a science and ... its significance has little to do with computers. The computer revolution is a revolution in the way we think and in the way we express what we think.

Harold Abelson and Gerald Sussman (1985), The Structure and Interpretation of Computer Programs, Cambridge, Mass.: MIT Press.

The activity of computer programming goes far beyond the laborious specification of detail, which it seems to be in the popular conception. Of course, there has to be careful attention to detail but the real challenge and difficulty lies in making explicit mental processes that heretofore have been implicit and in combining this with a multitude of other skills:

In my opinion programming is ... the most humanly difficult of all professions involving numbers of men ... the programmer is challenged to combine, with the ability of a first-class mathematician to deal in logical abstractions, a more practical, a more Edisonian talent, enabling him to build useful engines out of zeros and ones, alone. He must join the accuracy of a bank clerk with the acumen of a scout, and to these add the powers of fantasy of an author of detective stories and the sober practicality of a businessman. To top all this off, he must have a taste for collective work and a feeling for the corporate interests of his employer.

Andrei P. Ershov (1972), Aesthetics and the human factor in programming, Communications of the ACM, 15, 501-505.

Programming is quite a challenge, then, and not just for men.

Usually, programmers fail to meet the challenge, for it is a fact of life that almost all programs that have been written have been riddled with bugs:

The bugs which you would fright me with I seek.

William Shakespeare, The Winter's Tale, 3, 2.

The experimental approach to debugging programs, as recommended by Shakespeare, has the problem that it can never be guaranteed that the last bug has been found. Instead, it is proposed that we somehow 'prove' that our programs do what we claim they do. Alan Turing had discussed the two methods – the theoretical and the experimental – long ago:

It is of course important that some efforts be made to verify the correctness of the assertions that are made about a routine. There are essentially two types of method available, the theoretical and the experimental. In the extreme form of the theoretical method a watertight mathematical proof is provided for the assertion. In the extreme form of the experimental method the routine is tried out on the machine with a variety of initial conditions and is pronounced fit if the assertions hold in each case. Both methods have their weaknesses.

Alan Turing (1951), Mark I Programming Manual.

As Turing anticipated, the theoretical method is based on specifying assertions, that is, statements saying what parts of the program are intended to accomplish and which may be shown to be true:

Effectiveness of assertion is the alpha and omega of style.

George Bernard Shaw (1903), Man and Superman, Cambridge, Mass.: The University Press.

Edsger Dijkstra, a refined Dutchman who objected to the undisciplined nature of the programming activity, argued that writing a program and demonstrating that it is correct with respect to some specification should be carried out concurrently:

Instead of first designing the program and then trying to prove its correctness, we develop correctness proof and program hand in hand.

Edsger Dijkstra (1976), A Discipline of Programming, Englewood Cliffs, N.J.: Prentice-Hall.

In his book, he commented that "none of the programs …, needless to say [although he said it anyway], has been tested on a machine".

A 'discipline of programming': what a sombre subject it is in danger of becoming! Thankfully, programming in AI is different. Most AI programs do not compute functions, for which the program proving techniques are most suitable – often they carry out some continuous activity, such as having a conversation in natural language.

In conventional programming a programmer needs to start out knowing or at least believing that he can precisely specify how the problem being addressed may be solved. An AI programmer rarely knows in advance how or even if his problem can be solved. The point of attempting to write the program is often to see to what extent it might be solvable:

Software engineering problems are a subset of AI problems; the subset of well-defined problems.

Derek Partridge (1986), Artificial Intelligence: Applications in the Future of Software Engineering, Chichester: Ellis Horwood.

'Software engineering' is an expression of wishful thinking devised for a conference in 1968 intended to draw attention to:

... the need for software manufacture to be based on the types of theoretical foundations and practical disciplines that are traditional in the established branches of engineering.

Peter Naur and Brian Randell, eds. (1968), Software Engineering, Report on a conference sponsored by the NATO Scientific Committee.

Whether or not this need has subsequently been met we leave others to judge. Our point is that software engineering is not, as Partridge asserts, a subset of

AI: AI is not concerned at all with 'well-defined problems' for which the application of standard techniques is expected to lead to a solution.

AI programming is inherently exploratory in nature. We rarely have specifications of what programs are supposed to do with respect to which we may try to prove our programs. Who could possibly write a formal specification of a program intended to, say, recognise human faces in photographs or discuss the symptoms of some bacterial infection? Most AI programmers do not find computer programming a mechanical chore but a continual, exhilarating source of insight providing an aesthetic experience:

> **The source of the exhilaration associated with computer programming is the continual unfolding within the mind and on the computer of mechanisms expressed as programs and the explosion of perception they generate.**
>
> *Alan J. Perlis (1985), foreword, Harold Abelson and Gerald Sussman, The Structure and Interpretation of Computer Programs, Cambridge, Mass.: MIT Press.*
>
> **The process of preparing programs for a digital computer is especially attractive because it not only can be economically and scientifically rewarding, it can also be an aesthetic experience much like composing poetry or music.**
>
> *Donald E. Knuth (1968), preface, The Art of Computer Programming, Vol. 1, Reading, Mass.: Addison-Wesley.*

In fact, since neither Alan Perlis nor Donald Knuth was a member of the AI community, this might be regarded as the general view of programming, contrary to the common conception. In 1966 Perlis was the first recipient of the Alan Turing Award, now given annually to honour a computer scientist. Knuth also received the Alan Turing Award, in 1974, mainly in recognition of his *The Art of Computer Programming*, one of *American Scientist*'s best twelve scientific books of the century – the 20th century, that is, for the book was begun in 1962. It has evolved into a seven-volume epic, with the first three volumes being duly published and the fourth currently planned for 2007, and so on, perhaps.

9. Computer games: "chess is not such a difficult game"

Not only can programming be "much like composing poetry or music", as Knuth says, but it can also be *about* composing poetry or music. Computers are not calculating machines if calculation (from the Latin word for a small stone used for counting) is considered to be concerned only with arithmetic. The operands with which computers work can be regarded by us as

representing a number (say, 120507) or a date (12 May 2007) or a word ('leg') or indeed anything for which we feel able to define appropriate operations. Charles Babbage and Ada Lovelace had realised this, when reflecting on the powers of the Analytical Engine:

> **Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.**
>
> *Ada Lovelace (1843), Translation and notes of a Sketch of the Analytical Engine invented by Charles Babbage, by Louis Menebrae, in Richard Taylor (ed.), Scientific Memoirs, Selections from the Transactions of Foreign Academies and Learned Societies and from Foreign Journals.*

Babbage himself had no taste for music but he did contemplate using his machines for playing games such as chess as a money-raising venture, although he eventually decided it would be too much of a distraction from his more serious aims.

Computer pioneers also toyed with programs to play games. Turing discussed the possibility of computers playing chess (in fact, he also combined his interest in chess with that in long-distance running to devise a form of interval training in which it was necessary to run around the house between moves). Claude Shannon, who in the 1940s had invented information theory, described the first attempt at a chess-playing program as part of an argument that computers can deal with any kind of symbol, not just numbers, and Allen Newell (of whom more shortly) used chess playing as a context to establish one of AI's central methodologies, the requirement to develop working implementations rather than, or as well as, armchair theories:

> **[A computer could] be adapted to work symbolically with elements representing words, propositions or other conceptual entities.**
>
> *Claude Shannon (1950), A chess-playing machine, Scientific American, 182.*
>
> **These mechanisms are so complicated that it is impossible to predict whether they will work. The justification of the present article is the intent to see if in fact an organized collection of rules of thumb can pull itself up by its bootstraps and learn to play good chess.**
>
> *Allen Newell (1955), The chess machine: an example of dealing with a complex task by adaption, Proceedings of Western Joint Computer Conference, 101-108, Institute of Radio Engineers: New York.*

It says something about the human spirit that the earliest uses of computers were to play games and to support the war effort.

Both have continued to be a focus for AI research. Chess, in particular, has been considered to encapsulate many of the problems that AI needs to address:

Over the centuries, [chess] has come to be regarded as the intellectual game par excellence, so complex is its nature and so varied are the positions that can arise even within a very few moves ... if we could write a computer program that could play good chess we could (presumably) use similar programming techniques to solve other problems in long-range planning. Is it any more difficult to win the World Chess Championship than it is to plan the year's budget for a nation or to solve a difficult diplomatic crisis with the flair of a Kissinger? I doubt it.

David Levy (1976), Chess and Computers, Woodland Hills, California: Computer Science Press.

Despite this prediction, Levy himself made a small fortune through betting that no computer could beat him. He proceeded to jeopardise this income by offering advice to the designers of chess programs:

My feeling is that a human world chess champion losing to a computer program in a serious match is a lot further away than I thought. Most people working on computer chess are working on the wrong lines. If more chess programmers studied the way human chess masters think and tried to emulate that to some extent, then I think they might get further.

David Levy (May 12 1984), comment in Los Angeles Times.

This raises a question ubiquitous in AI: to what extent must or should a program to solve a problem emulate the way humans solve such a problem?

In May 1997 the chess program Deep Blue defeated the world chess champion Garry Kasparov 3.5 to 2.5. This contest was not played under championship conditions but we are clearly closer to the world champion chess program that Levy predicted than we are to computer-based diplomacy. The 2002 match between the program Deep Fritz and the world champion Vladimir Kramnik was a 4-4 draw. Perhaps chess is not such a reliable touchstone of intelligence after all as even cheaply available chess programs can already defeat 99.9% of the human chess-playing population and, of course, 100% of the rest (which is most of us):

"And can it really think?" asked one of them.

"It can beat me at chess," said Pender.

"Really?" said the man. "That's clever of it. But perhaps chess is not such a difficult game, if you don't play it the way the professionals do."

Lord Dunsany (1951), The Last Revolution, London: Jerrolds.

Edward John Moreton Drax Plunkett (1878-1957), the 18th Baron Dunsany, was a big game hunter, chess-master, soldier, and an influential fantasy

writer. *The Last Revolution* explored the familiar theme of machinery rising up against humanity. In his last years, Lord Dunsany had a protracted correspondence with a young Arthur C. Clarke, later, of course, to become a renowned visionary and probably best known for his *2001: A Space Odyssey* (1968), the most famous illustration of human-computer conflict.

Beating the world chess champion is, however, undeniably impressive. How was it achieved? It seems that the most significant factor was not the development of intelligent techniques for planning and decision-making based upon the thinking of human chess masters, as Levy advised. Rather, it was the phenomenal increase in computer speed that enabled Deep Blue to explore possible sequences of moves to a much greater depth. In the general view, this immediately disqualified Deep Blue's achievement from being considered a success for AI – in fact, quite the opposite:

> **Far from representing a triumph for the computer and artificial intelligence, the recent contest between Deep Blue and Garry Kasparov shows the poverty of intellectual capacity of a machine which needs to process 200 million possible chess moves a second in order to win a match against a man who can think about only three or four.**
>
> *Alan Fraser (May 13 1997), letter to The Times.*

This conclusion, dispiriting as it may be to AIers who had assumed that success at computer chess demanded advanced reasoning, intuition and creativity rather than brute force search, was conceded by experts in the field seeking to explain Deep Blue's success:

> **I consider the most important trend was that computers got considerably faster in these last 50 years. In this process, we found that many things for which we had at best anthropomorphic solutions, which in many cases failed to capture the real gist of a human's method, could be done by more brute-forcish methods that merely enumerated until a satisfactory solution was found. If this is heresy, so be it.**
>
> *Hans Berliner (2000), in Marti Hearst and Haym Hirsh (eds.), AI's greatest trends and controversies, IEEE Intelligent Systems, 15, 9.*

The significance of the Deep Blue versus Kasparov contest has been further clouded by the allegation that it was heavily biased against Kasparov as part of an IBM marketing strategy.

The first non-human world champion was a program called Chinook, which won the 1994 World Man-Machine Championship for checkers (draughts) when the long-standing champion Marion Tinsley withdrew through ill-health. Shortly after Deep Blue's achievement in 1997, the program Logistello beat the human world othello champion. However, there

is no sign yet of a world champion program for the games of shogi (Japanese chess) or go.

All these games might be considered 'easy' ones in the sense that players have access to all the information about the state of the game. Among games requiring some kind of inference about hidden information, backgammon and scrabble have world-champion level programs, but poker and bridge do not. Recently, the market of interactive computer games has provided a new impetus for AI-related game-playing research:

> **Computer games are the ideal application for developing human-level AI. There is already a need for it, since human game players are generally dissatisfied with computer characters. The characters are shallow, too easy to predict, and, all too often, exhibit artificial stupidity rather than artificial intelligence.**
>
> *Jonathan Schaeffer and H. Jaap van den Herik (2002), Games, computers, and artificial intelligence, Artificial Intelligence, 134, 1-7.*

Generally, though, multimedia gadgetry has more sales appeal than intelligence but at least the enterprise has demonstrated that computers are general-purpose symbol-processing machines.

But what's new? The mechanical engineers of the nineteenth century had already realised that machines can deal with symbols other than numbers:

> **[The Eureka] is the name of a machine for composing hexameter Latin verses, which is now exhibited at the Egyptian Hall, in Piccadilly. It was designed and constructed at Bridgwater, in Somersetshire; was begun in 1830, and completed in 1843 ... The rate of composition is about one verse per minute … it may be made to go on continually, producing in one day and night, or twenty-four hours, about 1440 Latin verses; or, in a whole week (Sundays included) about 10,000. During the composition of each line, a cylinder in the interior of the machine performs the National Anthem.**
>
> *Illustrated London News (1845).*

10. Symbols: "comfort and inspiration"

Two things appear to be new. First, the concept of a 'symbol' is enriched. In this context, a symbol is an entity with the property that when a computational process has a token of a symbol it has access to information about what that symbol designates (encoded in other symbolic expressions). Processes can create and delete symbols and can change the form and content of symbols. Hence, symbols do not have to be 'atomic' and with standard interpretations. Symbols can be accumulated into progressively more complex symbols and

computers can be thought of, at least, as dealing directly with the higher order symbols. For example, an atomic symbol in music might be a single note and such symbols may be gathered into higher order symbols such as a bar, phrase, minuet, and so on. Just as a competent musician thinks directly in terms of the appropriate higher order symbols and literally does not see the atomic symbols, so, in principle, could a programmer of computer music create and deal directly with these same higher order symbols. Moreover, the 'meaning' that the programmer arranges to be associated with her symbols is entirely up to her:

> **A person gets from a symbol the meaning he puts into it, and what is one man's comfort and inspiration is another's jest and scorn.**
>
> *Justice Robert Jackson (1943), West Virginia State Board v Barmotte.*

This, of course, is a mixed blessing – it provides freedom of expression but also scope for misinterpretation.

The word 'symbol' has a number of connotations, which has caused confusion. The word comes from the Greek 'sun ballo', to put together, and originally meant one part of something (such as a ticket) cut in two, which was presented to show a claim to the other part. The word 'symbol' retains some of this sense in linguistics and psychology, where a symbol, for example, the word 'donkey', is considered to, in the absence of a real donkey, trigger concepts concerning donkey-hood in the mind of the reader or hearer. What is triggered is more than just an image of a typical donkey and will be different for each of us. For less obviously referential words, such as 'honesty', it is much less clear what is triggered. In science, however, a symbol (for example, the 'c' in $e = mc^2$) stands for some entity in the subject domain, without any psychological associations. In mathematics a symbol sometimes (for example, the 'x' in $x + y = 3$) stands for anything at all, provided that it is constrained by the formula. In AI, a 'symbol' can have aspects of all these (and other) natures.

The second new thing is the view that it suggests about human thinking. When we describe the operation of the human eye by analogy to the camera we appreciate that this is a very superficial metaphor. However, when some researchers in AI compare human thinking with computer symbol-processing they are explicitly not being metaphorical: they mean that, in their opinion, human thinking really does consist of processing symbols in much the way that computers do.

For this identification of human thinking with symbol-processing to be useful we need a precise definition of the latter. Allen Newell and Herbert Simon, the original and strongest proponents of this view, developed what they called the 'physical symbol system hypothesis' as follows:

> **A physical symbol system consists of a set of entities, called symbols, which are physical patterns that can occur as components of another type of entity called an expression (or symbol structure) ... At any instant of time the system will contain a collection of these symbol structures ... A physical symbol system is a machine that produces through time an evolving collection of symbol structures ... [such a] system has the necessary and sufficient means for general intelligent action.**
>
> *Allen Newell and Herbert Simon (1976), Computer science as empirical enquiry: symbols and search, Communications of the ACM, 19, 113-126.*

This hypothesis is more one of psychology than it is of computation. If computers are accepted to be symbol-processing machines then if any intelligent action is forthcoming from a computer then it must be due to the symbol system within the computer. The novelty lies in the proposal that a symbol system is necessary and sufficient for any intelligent action, including that *by humans*.

A related hypothesis, the 'knowledge representation hypothesis', considers the requirements of an intelligent machine:

> **Any mechanically embodied intelligent process will be comprised of structural ingredients that (a) we as external observers naturally take to represent a propositional account of the knowledge that the overall process exhibits, and (b) independent of such external semantical attribution, play a formal but causal and essential role in engendering the behavior that manifests that knowledge.**
>
> *Brian Cantwell Smith (1982), Reflection and semantics in a procedural language, PhD thesis (Technical Report No. 272), MIT Laboratory for Computer Science.*

In other words, according to the hypothesis, an intelligent machine must have what we consider to be a representation of knowledge that is used to generate the machine's behaviour.

At the time these two hypotheses were verbalised they were not seen as surprising. Rather, they were an attempt to summarise and consolidate the outcomes of the first generation of AI research. The hypotheses are not amenable to mathematical or logical proof. The degree to which they hold can only be determined by empirical investigation, allowing for some subjectivity in the meanings of phrases such as 'physical patterns', 'general intelligent action', 'causal and essential role' and even 'symbol'. At all events, a key consideration in determining the significance of AI concerns whether system developers are seeking a non-metaphorical description of human thinking, or are only interested in achieving some level of performance. When we see an AI program perform an apparently intelligent action, say, solving a crossword, to what extent do we consider that the way

the program works is or must be similar to the way that the only other entity that can solve crosswords works?

Of course, any analogy does not lie at the physical level:

> **I find it quite amazing that it is possible to predict what will happen by ... simply following rules which really have nothing to do with what is going on in the original thing. The closing and opening of switches in a computer is quite different from what is happening in nature.**
>
> *Richard Feynman (1965), The Character of Physical Law, New York: Random House.*

If there is an analogy or equivalence it lies at what Newell and Simon call the symbol-processing level. When an AI programmer describes the symbols and structures used in, for example, his crossword solving program, it is tempting to interpret these as a proposal for the way we solve crosswords: surely the human mind must also use such symbols and structures to solve crosswords.

Consider, however, a different kind of theory. We can write equations based on Kepler's theory of planetary motion to predict the motion of Neptune. The equations are a representational convenience for us: they enable us to understand and reason with the theory. Nobody believes that, because the behaviour predicted by the equations is exactly that observed of Neptune, Neptune must also 'possess' those equations and must be executing them in some sense. With a computer program simulating human thought processes, however, we are led to make just such an inference – that the human brain must also possess symbols and be processing them in a similar way to the program. It is Newell and Simon's thesis that this is indeed the case, and we will return to consider this later.

The notion that human behaviour may be governed by rules, which we can simulate with symbol-processing operations in computers, may not raise many eyebrows today but it should be put in its historical context. 'Rule-governed behaviour' was very much in the air at the time it came to AI, in the 1950s and 1960s. Noam Chomsky was at that time revolutionising linguistics with his work on syntactic structures and transformational grammars, in which explicit rules describe a language and can be used to generate sentences in that language. In music, composers such as Pierre Boulez and Iannis Xenakis were experimenting with the rule-based generation of music, sometimes modified by random elements. Earlier composers, such as Bach with *The Art of Fugue*, had demonstrated the formal manipulation of musical material but it was only with the advent of the computer, able to execute the rules specified, that it became a serious mode of composition. (The conventional term 'execute' is unfortunate in this context: 'animate' would

be better.) Even in art, explicit rules were being developed to generate paintings in the style of, say, Kandinsky or Miró, and in architectural design shape-grammars were being developed to generate houses in the style of Frank Lloyd Wright, for example. Jean Piaget also applied such structuralist methods to formalise his ideas about developmental psychology.

Previously, it had largely been assumed that all these activities had an abstract, ineffable quality in which meaning derived somehow from the interrelationships within the systems. Once rule-governed behaviour became an established paradigm it then became adopted, almost without question, for other activities. For example, researchers began writing about the 'grammar of vision', whereby visual perception involves 'parsing' the sensory input into meaningful, ambiguous or impossible 'sentences'.

11. Structures: "a mighty sense of accomplishment"

How, then, may we define and process complex symbols within computers? If we consider the case of music, we might say that a 'chord' consists of a number of notes, a 'bar' consists of a number of chords, and so on. Chords might also have dynamic and timbre features; bars might also have tempi. We would include such features in our definitions if our problem required us to take them into account. In general, we need to be able to say, for each symbol, what kinds of sub-symbol it consists of.

Modern programming languages provide ways of defining such data structures, as they are called. We can see that the general-purpose, abstract nature of symbol-processing makes the computer far from the old view of a machine as a device transmitting or modifying force:

> **The protean nature of the computer is such that it can act like a machine or like a language to be shaped and exploited. It is a medium that can dynamically simulate the details of any other medium, including media that cannot exist physically … It is the first metamedium, and as such it has degrees of freedom for representation and expression never before encountered and as yet barely investigated.**
>
> *Alan Kay (1984), Computer software, Scientific American, 251, 3, 52-59.*

After developing the concept of a personal, portable computer in the early 1970s, some twenty years before they were practical, Alan Kay became acknowledged as a computer guru, leading to, for example, an appointment as Disney Fellow with the Walt Disney Company.

If we were considering a computer program to deal with language, we might use a 'word' symbol consisting of letters and a 'sentence' consisting of words, and so on. This, however, is not a deep analysis – the words in a

sentence are organised in ways to communicate meaning, and if our problem concerns the sentence's meaning then we should try to capture this organisation somehow (but not otherwise – for example, if we just wish to count the occurrences of different words). As we may recall from school parsing exercises, sentences might be considered to consist of 'noun phrases', 'verb phrases', 'predicates', and so on. We might represent the structure of a sentence by writing 'sentence' at the top of a page, its main constituents immediately below it, the constituents of those constituents below them, and so on, down to, at the bottom of the page, the actual words in the sentence. Such a representation is called a 'tree':

I think that I shall never see a billboard lovely as a tree.

Perhaps, unless the billboards fall, I'll never see a tree at all.

Ogden Nash (1933), Song of the Open Road.

This ode is a parody of the well-known 1913 poem of Joyce Kilmer, which compared a poem to a tree.

Lovely as it may be, our tree is rather unusual because the top-most position is the 'root' of the tree, going down through the 'branches' to the 'leaves':

There is a tree, the tree of Transmigration, the Asvattha tree everlasting. Its roots are above in the Highest, and its branches are here below. Its leaves are sacred songs, and he who knows them knows the Vedas.

Bhagavad Gita, Chapter 15.

The familiar family tree of descendants has a name of a person at the root and below that the person's children and then their children and so on. A family tree of antecedents has the name, then the parents, then the grandparents, and so on. A tree with at most two branches from each node is called a 'binary tree'.

In the case of chess, we might imagine writing down a board position at the top of the page, immediately below it all possible positions reachable by one move, below each of those positions all possible positions reachable by the opponent's move, and so on, down to board positions which are end-of-game positions. We had better imagine it rather than do it because in this case the process, like the Asvattha tree, would be everlasting.

There is, however, a possible complication. The symbol ('board position') is composed of symbols of the same kind (that is, other board positions). The values of those symbols will differ but their structures are the same. This seems to present a problem: how can we define to a computer the structure of a symbol whose components are, at the time of definition, undefined, since they are the very symbols we are in the process of defining?

The answer lies in the distinction between a village and a signpost to the village. The village of Reepham contains signposts to the villages of Guestwick, Bawdeswell and Alderford. It obviously does not contain the villages Guestwick, Bawdeswell and Alderford. To define the village of Reepham we could say it consisted of (amongst other things) three 'signposts to villages', not three 'villages'. Similarly, a 'board position' is composed of so many 'signposts (or "pointers", we usually say) to board positions', not the board positions themselves. In computing terms, the 'board position' symbol contains pointers to board positions, that is, something that tells the computer where these board positions are – or will be, since we can, if we wish, put up a signpost to a village before the village has actually been built. Therefore, we do not, as programmers, need to work out in advance all the details of the 'board position' data structure: we can leave the computer to work out all, or as many as it needs, of the succeeding board positions when they are required.

There is no reason at all that a symbol should always point to a symbol of the same type (a signpost can point to an airport, a rubbish tip, as we wish). So we can define data structures of whatever complexity we desire, and moreover (since the computer can change the values of pointers, just like anything else) our computer programs can build and modify such structures:

> **It was truly a splendid structure, and Yossarian throbbed with a mighty sense of accomplishment each time he gazed at it and reflected that none of the work that had gone into it was his.**
>
> *Joseph Heller (1961), Catch-22, New York: Simon and Schuster.*

These data structures are sometimes called 'information structures' and hence computers are called 'information processing' machines.

The term 'information' is here used in a technical sense and may not correspond to the everyday intuition of what the word 'information' means:

> **Consider the total quantity of information available to a talented person without a computer. Suppose this individual has completed a speed reading course and can read 1000 words per minute. One word is about five letters or 25 bits, so if this person spends six hours a day reading, seven days a week, for seventy years, he will have read about 2×10^{11} bits. A modest home computer can read that many bits in a few days – a really fast computer, in minutes. Therefore even today's computer technology provides an increase of a factor of thousands to millions in total information availability.**
>
> *Douglas Robertson (1998), The New Renaissance: Computers and the Next Level of Civilization, Oxford: Oxford University Press.*

Here, not only is the word 'information' misused but also the word 'read'. A computer 'reads' in the technical sense of transferring bits of information from one place to another, which is far from the everyday sense of reading. This is a perpetual problem. Computer science, and AI especially, often uses everyday words as technical terms (because of a rough analogy) rather than inventing new words, which would make computing literature even more inscrutable than it is. We need to be very wary about ascribing more to a word than is really there.

Dynamic data structures, that is, ones that are built by the computer program while it is being executed, are crucial for AI programming. They are the means for describing the knowledge that programs need to solve problems and for enabling programs to reason about the knowledge itself. In AI, the basic data structure is a list, which is essentially a binary tree. The programming language Lisp, for list processing, was devised by John McCarthy in 1958 and, astonishingly, is still the most widely used language for AI. Lisp was designed for symbolic, rather than the then-dominant numeric, computations and, uniquely for the time, was based on a formal model of computation, the theory of recursive functions (devised by Kurt Gödel in the 1930s and shown to provide a class of functions equivalent to those computable by Turing machines). A small set of primitive functions (analogous to the basic functions of arithmetic) is provided to operate upon lists of symbols and in terms of which programmers define higher-level functions. Programs are written as the nested application of functions:

```
largest-of(satellites-of(largest-of(planets-of(sun))))
```

These functions are necessarily evaluated from the inside out, giving as intermediate results a list of the planets, then Jupiter, then a list of its satellites, and finally Ganymede.

New functions are defined using a similar notation. Imagine that you have an AI glossary, giving a list of AI technical terms and their definitions, such as

(A*, a form of heuristic search), ..,
(heuristic, device intended to limit search), ..,
(nodes, points in a graph connected to other points), ..,
(optimal, generates the fewest nodes in finding a solution), ..,
(recursion, defining something in terms of itself), ..,
(search, process of looking through the set of possible solutions), ..

Now, let us say that the meaning of an 'AI sentence' is given by replacing all the words in the sentence that occur in the glossary by their definitions (and leaving a word not in the glossary unchanged).

Such a function might be defined by:

```
meaning(sentence) =
   if single-word(sentence) then
      if in-glossary(sentence)
         then definition(sentence) else sentence
   else join-together(
            definition(first-word(sentence)),
            meaning(rest(sentence)))
```

So, for example,

```
meaning(A* may be optimal)
```

returns the result:

```
a form of heuristic search may be generates the fewest nodes
   in finding a solution
```

But wait – some of the words in this result are also in the glossary. To obtain a complete result we should replace those words by their meanings too, and so on. This may be achieved by changing the last three lines of the above definition to:

```
   else join-together(
            meaning(definition(first-word(sentence))),
            meaning(rest(sentence)))
```

yielding (assuming none of these words is in the glossary):

```
a form of device intended to limit process of looking
   through the set of possible solutions process of looking
   through the set of possible solutions may be generates
   the fewest points in a graph connected to other points in
   finding a solution
```

Perhaps we were a little optimistic in calling our function '`meaning`'.

Our function definition is not in Lisp syntax but it gives something of the style of Lisp. As may be seen, Lisp programs rely heavily on the use of recursion. The syntax is the same for data and programs, and so, just as it is possible to manipulate data, we can define and manipulate programs as data and hence effectively define new languages appropriate to the problem at hand. The elegance of the basic design has meant that it has been possible to keep extending the language to retain its leading position in AI. Apart from providing powerful ways of defining new symbols and functions, the modern Lisp programming environment (with incremental compilers, debugging tools and interactive interfaces) facilitates the fast, experimental development of programs.

AI programming, then, differs from conventional software engineering in involving a form of qualitative modelling (as opposed to quantitative or numeric representations) in which relational networks, representing causal,

spatial, temporal and other relations, are constructed and manipulated by the systems themselves (as opposed to being described in detail by programmers). For example, we could define a network representing the relationships between symptoms, diseases and treatments, as a way of modelling the health of a patient. This contribution to computing methodologies might come to be seen as AI's most important one:

> **This generation of AI research may come to be viewed not so much for the particular capabilities of the programs that were developed, but for the generality of the methods, which as computational formalisms are arguably as novel and wide-sweeping in their impact as Newton's calculus.**
>
> *William Clancey (1992), Model construction operators, Artificial Intelligence, 53, 1-115.*

12. Search: "a core area of AI"

If we, or rather our programs, are to build such complex structures then clearly we need to develop methods for our programs to find their way about them. These searches are characteristic of AI and in the early days were often taken to be definitive of AI:

> **... many of the problems that fall within the purview of artificial intelligence are too complex to be solvable by direct techniques; rather they must be attacked by appropriate search methods armed with whatever direct techniques are available to guide the search.**
>
> *Elaine Rich (1983), Artificial Intelligence, New York: McGraw-Hill.*

A distinction is drawn between conventional programming and AI programmer: in the former case, the programmer herself analyses the problem and then specifies a sequence of steps (an algorithm, from the ninth century Persian mathematician, abu-Jafar Mohammed ibn-Musa al-Khuwarizmi) for the computer to carry out to provide a solution; in the latter case, where the programmer is unable to specify an algorithm, because the problem is too hard, she instead provides a general problem solving method, together perhaps with whatever guidelines and constraints she can think of, and then sets the computer to search for a solution. An AI 'search' may be more or less systematic.

Returning now to our problem of unjumbling a Rubik cube, our program could build a tree, containing at the root the initial jumbled state, then all the states you could reach by a single twist, then all the states you could reach by a single twist applied to those states, and so on. If the program ever generates a state that is your 'goal', the sorted state, then it has found a sequence of twists that can unjumble the cube:

He that would have the fruit must climb the tree.

Thomas Fuller (1732), Gnomologia: Adages and Proverbs.

This, incidentally, is the Thomas Fuller (1654-1734) whose proverbs were plagiarised by Benjamin Franklin, not the Thomas Fuller (1710-1790) who was an African slave shipped to the United States in 1724 and, although unable to read or write, became renowned for his extraordinary powers of mental arithmetic.

Of course, the program is not physically operating on the cube – it is carrying out the twists 'in its mind':

Know that I have gone many ways wandering in thought.

Sophocles (427 B.C.), Oedipus the King.

Clearly, the program will get in a twist, as we do, unless it is careful in generating new states. There are two basic methods. It could generate the branches level by level from the root, that is, all the states reachable by one twist, then all those by two twists, then all those by three twists, and so on. This is called a 'breadth-first search' and is guaranteed to give a solution with the minimum number of twists. Unfortunately, it requires a lot of computer memory, since it is necessary to remember all the states reachable by n twists in order to generate those reachable by n+1 twists.

The other basic method (a 'depth-first search') involves applying a single twist to generate a new state, then (ignoring any alternative first twists) taking that state to generate a new state, then (ignoring any alternative second twists) taking that state to generate a new state, and so on:

A straight path never leads anywhere except to the objective.

André Gide (1922), Journals.

Sorry, but this method is not guaranteed to lead to the objective at all. It is easy to imagine applying inappropriate twists and keeping the cube perpetually jumbled. With some problems, though not with the Rubik cube, we may just run out of new things to try. So, to get out of never-ends and dead-ends, our program needs to be able to return to previously ignored alternatives, by systematically undoing its choices:

A road that does not lead to other roads always has to be retraced, unless the traveller chooses to rust at the end of it.

Tehyi Hsieh (1948), Chinese Epigrams Inside Out and Proverbs.

If these alternatives turn out to be no better, our program could return along this abandoned path. Backtracking in this way also overcomes the potential difficulty of being stuck in a loop when a sequence of twists generates a state the program has generated before:

If he knew where he was going, it is not apparent from this distance. He fell down a great deal during this period, because of a trick he had of walking into himself.

James Thurber (1945), The Thurber Carnival, preface, New York: Harper Collins.

This is a source of inconvenience and inefficiency in finding a solution and, in general, can only be avoided by the program keeping some record of previously generated states, which is, of course, demanding of computer memory. Even if our program succeeds in finding a solution by such a depth-first search we have no guarantee that it is the *best* solution (that is, the shortest, in this case). There may be a better solution using an ignored alternative.

Both these basic methods, the breadth-first search and the depth-first search, are exhaustive searches in that all possible states are generated in due course and consequently if there is a solution they are bound to find it – eventually. But in general for any problem of interest it takes much too long to carry out an exhaustive search.

There are various things we could try in order to speed things up. We could look for properties of the search that we could capitalise on. For example, with our Rubik cube, the initial state and the goal state are symmetrical. We could, if we wished, arrange to search from the goal state, imagine applying twists and hope to generate the jumbled state – reversing the twists used to get the jumbled state would give a solution to the original problem:

We often get in quicker by the back door than by the front.

Napoleon I (1804-15), Maxims.

For some problems it is easier to work backwards from the goal state. For our Rubik cube problem, it generally makes no difference. We could generate the two searches in parallel, from the initial and goal states, until we found a state common to the two searches. That would decrease the number of states generated by about a half, in fact, for our Rubik cube problem.

Or we could think about the way we describe the state and the operations on it. The way we represent the states as computer data structures can radically influence the size of the search space. Also, expert Rubik cube solvers might not think in terms of single twists but in terms of 'macro-operations' such as getting a particular row of faces all the same colour. If we generated states using such macro-operators then we might hope that the number of states generated would decrease:

A fool sees not the same tree that a wise man sees.

William Blake (1790), The Marriage of Heaven and Hell.

Possibly, Blake was referring to an incident when, at the age of nine, he had alarmed his parents by saying that he had seen a tree full of angels, although it is unclear which of the young Blake and the parents constitute the fool and the wise man. When he died in 1827 Blake was generally thought to have been insane but a reassessment has led to him being regarded as a seer warning of the dangers of a world perceived as mechanism, with man a mere cog.

Sometimes, instead of applying operations at random, it is possible to order the potential operations in terms of their perceived promise of leading to a solution. For example, if your problem is to travel from Paris to Los Angeles then it might be thought more promising to take a flight to New York than one to Amsterdam. But it is not necessarily the case. It is in general very difficult to define what are called 'evaluation functions' which are able to judge whether intermediate states are promising avenues to the desired goal state:

> **All the time life is a fork. If you are straight up with yourself you don't have to decide which road to take. Your karma will look after that.**
>
> *George Harrison (c1970).*

Alas, no AI programmer has yet managed to convert the notion of karma into an effective procedure.

A device to limit the search space by avoiding the need for an exhaustive search is called a 'heuristic':

> **Heuristic search remains as a core area of artificial intelligence. The use of a good search algorithm is often a crucial factor in the performance of an intelligent system.**
>
> *Weixiong Zhang, Rina Dechter and Richard Korf (2001), Heuristic search in artificial intelligence, Artificial Intelligence, 129, 1-4.*

Heuristics are generally not guaranteed to find an optimal, correct solution but they are intended to help find useful solutions more efficiently. Sometimes heuristics introduced to improve efficiency have been refined to regain the property of providing optimal solutions and sometimes heuristics can be shown to provide the optimal solution to any desired degree of approximation. Usually, however, heuristics are just devices to limit otherwise intractable searches, which will always be necessary in AI because of the complexity of the problems being addressed.

However, any move towards evaluation functions, specialised operators or state descriptions is a move away from generality. Our program is now tainted with some knowledge that ***we*** possess that is specific to the problem at hand. If this is allowed, then we could include all sorts of guidelines, rules of thumb, and constraints that might help the program select the most

promising states to generate from first. Of course, it is allowed if our aim is just to solve a specific problem, for it is rare indeed that we know nothing that can help a search for a solution to a problem. However, AI has been reluctant to give up theoretically pure general problem solving, as we'll see.

13. Problem solving: "weak and shallow"

For the moment, let us concentrate on the view of problem solving that is being made explicit. The idea is that we describe a problem in terms of the transformation of some initial state into a state that meets the conditions required of a solution to the problem (such as 'the opponent is in a checkmate position') using a sequence of operators selected from some set. Problem solving involves searching this space of states:

> **When solving problems ... one will probably apply some very complex "transformation" of the original problem, involving searching through various variables, some more analogous to the original one, some more like a "search through all proofs." Further research into the intelligence of machinery will probably be very greatly concerned with "searches" of this kind.**
>
> *Alan Turing (1948), Intelligent machinery, first published in Bernard Meltzer and Donald Michie, eds. (1969), Machine Intelligence 5, Edinburgh: Edinburgh University Press..*

> **One of the most widely-used frameworks for problem solving in artificial intelligence is state-space search.**
>
> *Eric Hansen and Shlomo Zilberstein (2001), LAO*: a heuristic search algorithm that finds solutions with loops, Artificial Intelligence, 129, 35-62.*

Thus, this view pre-dates the subject of AI itself but is still thought central to AI. It can be seen as an attempt to pass on to the computer the onus for problem decomposition (using subroutines) that we had previously assigned to programmers.

The system designer, therefore, has to define four things:

- A way of describing the states (including the initial state).
- The operations or actions that can be carried out to transform one state into another (and the conditions under which they may be applied).
- The goal, which is a condition that can be applied to a state to see if a solution has been found.
- A 'cost function', which judges the cost of a sequence of actions (often this is just the number of actions but sometimes some actions are most costly in some sense than others, for example, they make take longer to carry out).

These four components are said to define a problem, with a solution being a sequence of actions that transforms the initial state into one satisfying the goal condition. If there were several possible solutions then we would prefer to find the one with minimum cost.

Sometimes it is helpful to view state-space search as 'problem reduction'. The state then describes the difference between the current situation and the desired one (that is, the problem) and actions transform problems into sub-problems. Sometimes a problem can be transformed into a number of sub-problems (rather than just one) and it is necessary to solve them all to consider the original problem solved.

For example, to send a letter (in the traditional way) it is necessary to write the letter, address an envelope, put a stamp on the envelope, enclose the letter in the envelope, find a post-box, and put the envelope in the post-box. Some of these sub-problems are independent of the others and so can be tackled separately; others depend on other sub-problems having already been solved. Formalising these kinds of inter-relationships is a continuing concern in automatic problem solving.

The degree of decomposition of the problem is a matter of choice. We might, for example, take 'prepare the envelope' as a sub-problem, which includes both addressing and stamping the envelope. And, as this implies, any sub-problem which is not directly solvable has to be decomposed into further sub-problems, until, it is hoped, solvable sub-problems are reached:

> **If we understand a problem perfectly, it should be considered apart from all superfluous concepts, reduced to its simplest form, and divided by enumeration into the smallest possible parts.**
>
> *René Descartes (1637), Rules for the Direction of the Mind.*
>
> **The mightiest rivers lose the force when split up into several streams.**
>
> *Ovid (c. 8 A.D.), Love's Cure.*

The basic idea, then, is to follow the traditional advice to break a difficult problem into a number of more manageable ones.

If a problem can be divided into a set of sub-problems, a natural question to arise is whether any sub-problems may be solved in parallel. If they are then in the case of a computer solution we imagine the sub-problems being delegated to different processors running in parallel. These processors might need similarly to delegate to further processors. When the processors have completed their sub-tasks they would communicate their solutions back to contribute to the original problem solution. A part of computer science is concerned with the possibilities of parallel processing. Although parallel processing has its place (for example, it played a crucial part in enabling the efficient search of board positions in Deep Blue), it is a complicated matter

and often it does not provide a means of obtaining solutions more efficiently because in most cases the solutions to sub-problems need to be communicated to other sub-problems before they may be tackled, so that, in effect, sub-problems have to be addressed serially:

Two roads diverged in a yellow wood,
And sorry I could not travel both.

Robert Frost (1920), The road not taken.

To do two things at once is to do neither.

Publilus Syrus (1st century B.C.), Moral Sayings.

The human difficulty in parallel processing applies to the task level – it is hard for us to solve a crossword and write a letter at the same time. The same difficulty may not apply to computers, for parallel processing can be simulated by software even without real parallel processors. At a lower level, however, it is different. The neurons of the brain, of which there are about 10^{11}, operate in parallel. Computer technology does not yet match this although researchers remain optimistic that it might.

Sometimes a sub-problem is of the same form as the original problem. For example, with our problem of unjumbling a Rubik cube a sub-problem is to unjumble a Rubik cube that is the starting cube after a single twist. The leads naturally to the use of recursive programs, as was mentioned before. Arguably the first genuine AI program, the Logic Theorist written in 1955 by Allen Newell, Herbert Simon and Cliff Shaw, used such methods to prove theorems in Whitehead and Russell's *Principia Mathematica* (1910). The program included a number of problem solving techniques that were refined and made central in its immediate successor, the General Problem Solver or, more acronymically, GPS.

In particular, GPS used the technique of 'means-end analysis', whereby the difference between the initial and goal states is computed and the initial state is recursively transformed by the application of operators which heuristic rules suggest will reduce this difference: a deceptively simple technique which Newell and Simon made formally precise and applied to a number of problems, such as logic, cryptarithmetic and Tower of Hanoi puzzles. Newell and Simon were understandably coy about their use of the word 'general'. They acknowledged that there might well be more to thinking than the narrow puzzle solving that GPS managed and their first paper on GPS referred to "pretences to generality". Nonetheless, the idea that there were general techniques that could be applied to all problems stimulated great interest, although today GPS is regarded as one of the first instances of the use of overly suggestive labels for an AI program and its processes:

Remember GPS? By now, "GPS" is a colorless term denoting a particularly stupid program to solve puzzles. But it originally meant "General Problem Solver," which caused everybody a lot of needless excitement and distraction. It should have been called LFGNS – "Local Feature-Guided Network Searcher".

Drew McDermott (1976), Artificial intelligence meets natural stupidity, SIGART Newsletter, 57.

While the legacy of GPS is still debated, the Newell and Simon partnership undoubtedly formed the foundation for the new subjects of cognitive psychology and artificial intelligence, although their backgrounds suggest that they saw the new subjects as branches of operations research, intended to aid complex decision-making in business and industry. At a conference in 1985 Newell was giving the keynote address and the chairman gave the introduction, extolling the virtues of the great man, ending with "… and here he is, Herb Simon". Unfazed, Newell began, "Everything you said is true of Herb too". They were not indistinguishable, however. Herbert Simon (1916-2001) had a broad interest in human decision-making and problem solving processes and the implications of these processes for social institutions, and applied this interest to psychology, administration and economics, as well as AI. In 1978 he was awarded the Alfred Nobel Memorial Prize in Economic Sciences and in 1986 the National Medal of Science. He received the Alan Turing Award in 1975, with Newell. Allen Newell (1927-1992), a colleague at Carnegie Mellon University from 1961, was awarded the National Medal of Science for his contributions to the design of computer operating systems and for his mission to develop a unified computational-psychological theory describing human behaviour.

GPS was renowned as the first system explicitly to model human problem solving and for the attempt to separate general problem solving methods from specific knowledge but it was criticised on two grounds: as an inadequate approach to the engineering of effective solutions to real problems and as an incomplete view of the nature of human problem solving:

GPS was a dream come false.

John Haugeland (1985), Artificial Intelligence: the Very Idea, Cambridge, Mass.: MIT Press.

In particular, there was a shift away from the quest for all-powerful search and reasoning methods, capable of being applied to any kind of problem, towards a recognition that such methods are inherently inefficient. Instead, it began to seem that the key lay in trying to bring to bear large amounts of knowledge specifically appropriate to the problem at hand.

Those researchers mainly interested in the development of computer systems to perform complex problem solving to achieve practically useful ends denied the importance of general problem solving skills, believing instead that the use of domain-specific knowledge was the central issue:

> **General problem-solvers are too weak to be used as the basis for building high-performance systems. The behaviour of the best general problem-solvers we know, human problem-solvers, is observed to be weak and shallow, except in the areas in which the human problem-solver is a specialist.**
>
> *Edward Feigenbaum, Bruce Buchanan and Joshua Lederberg (1971), On generality and problem solving: a case study using the DENDRAL program, in Machine Intelligence 6, Edinburgh: Edinburgh University Press.*
>
> **The fundamental problem of understanding intelligence is not the identification of a few powerful techniques, but rather the question of how to represent large amounts of knowledge in a fashion that permits their effective use and interaction.**
>
> *Ira Goldstein and Seymour Papert (1977), Artificial intelligence, language, and the study of knowledge, Cognitive Science, 1, 84-123.*

This is a debate to which we will return.

14. Representation: "grounded in the physical world"

The apparently clinical approach to the messy world of human problems was anathema to many. The everyday use of the word 'problem' does not bring to mind examples that map neatly onto ideas of states, operators, goals and cost functions. In many cases, the real problem seems to lie in defining or understanding the problem itself – once it becomes expressible in formal terms then its solution may be relatively straightforward:

> **Once an appropriate representation is available, many problems do become amenable to automatic solution. In our view, however, the problem requiring intelligence is the original one of finding a representation. To place this problem in the domain of the system designer rather than that of the designed system is to beg the question and reduce intelligence to symbol manipulation.**
>
> *George Reeke and Gerald Edelman (1988), Real brains and AI, Daedalus, Winter, 143-73.*
>
> **If we really understand the problem, the answer will come out of it, because the answer is not separate from the problem.**
>
> *Jiddu Krishnamurti (1970), Questions and Answers, in The Krishnamurti Reader, London: Penguin.*

It isn't that they can't see the solution. It is that they can't see the problem.

G.K. Chesterton (1935), The Scandal of Father Brown, New York: Dodd, Mead & Co.

I have yet to see any problem, however complicated, which, when you looked at it in the right way, did not become still more complicated.

Poul Anderson (1969), New Scientist, Sept 25.

If a computer is a device for manipulating symbolic representations and it is somehow begging the question of what intelligence is to require system designers to provide the required representations, what alternative is there?

There seem to be only two possibilities. One is that the system somehow creates or discovers the representation itself, and we will consider this possibility later. The other is that a system does not need representations at all. Instead of developing an internal, computational representation of the physical world (even for something as artificial as a Rubik cube) we could provide the system with access to the physical world, so that it serves as its own representation, in a way:

The symbol system hypothesis upon which classical AI is based is fundamentally flawed ... To build a system that is intelligent, it is necessary to have representation grounded in the physical world … At each step we should build complete intelligent systems that we let loose in the real world with real sensing and real action.

Rodney Brooks (1991), Intelligence without representation, Artificial Intelligence, 47, 139-159.

Brooks challenged the conventional wisdom of the 'deliberative' approach to artificial intelligence. Instead of systems serially representing, reasoning, planning and then acting in the world, he designed 'reactive' robots in which a simple parallel, distributed control mechanism managed a set of independent active modules. W. Edwards Deming (1900-93), considered to be the father of the Japanese post-war industrial revival and recipient of the US National Medal of Technology in 1987, identified one aspect of the distinction:

Rational behavior requires theory. Reactive behavior requires only reflex action.

W. Edwards Deming.

For AI, the idea is hardly new: the cybernetic robots that pre-dated AI could be considered to be reactive robots. With reactive robots, there is minimal communication between the parallel controllers, with the system relying on the environment to serve as an external memory. How far such reactive robots can go without developing the need for symbolic representations remains a matter of controversy. The methodology of 'behaviour-based AI'

places robotics at the centre of AI, rather than, as it often appears, on the frivolous fringe, because, according to its proponents, intelligence is just not possible without embodiment.

Apart from the difficulty of developing representations that the reactive approach considers unnecessary, there are other complications for the deliberative approach. First, some of the actions decided upon are not to be carried out immediately: the outcomes of reasoning and planning may be a commitment to act at some future time. But the reactive approach doesn't address this difficulty at all and hardly seems to do justice to the view that long-term planning is a distinctive human capacity:

> **It is precisely this unique human capacity to transcend the present, to live one's life by purposes stretching into the future – to live not at the mercy of the world, but as a builder and designer of that world – that is the distinction between human and animal behavior, or between the human being and the machine.**
>
> *Betty Freidan (1963), The Feminine Mystique, New York: W.W. Norton.*

Secondly, the deliberative approach tends to assume that reasoning and planning is sufficiently instantaneous that the representation of the world does not become out of date, thereby invalidating the results of those processes. And thirdly, it assumes that the process of acting does not itself change something that affects the appropriateness of the actions being carried out.

Reasoning about the world is difficult but when the world of interest is an artificial one like a chessboard or an algebra problem then it is easy to imagine that a computer can be provided with a complete description of this world with which to reason. If, however, a computer robot were working in, say, an office, then this would be infeasible, even for an office as orderly as mine. Moreover, it may be unnecessary. For example, if the robot were to use a calculator (unlikely, admittedly) then it would not need a description of the layout of its keys because it could simply look at the calculator when it needs to, as we do. This knowledge is never needed except in the context of calculator use, so it always to hand when it is needed. The robot can use the world itself as a resource, without needing to describe it. However, acknowledging that some knowledge may be 'to hand' in the environment is not to say that all of it, or even much of it, can be. What, in the world, would tell our robot how to use the calculator to work out, say, a correlation coefficient, if there is no key for that? Also, it would not seem very intelligent if a robot had repeatedly to go and sense the environment in order to answer the same question.

15. Plans: "gangaftagley"

In practice, we tend not to delegate our problems to experts or computers for them to solve on our behalf. We prefer to be interactively involved in discussing the problem and collaborating in developing a solution. Recently, there has been more emphasis on the design of productive human-computer collaborative problem solving, but the method has its difficulties:

> **"... I want to work this out. Computer!"**
>
> **"Hi there!" it said brightly...**
>
> **"Oh God," said Zaphod. He hadn't worked with this computer for long but had already learned to loathe it.**
>
> **The computer continued, brash and cheery as if it were selling detergent. "I want you to know that whatever your problem, I am here to help you solve it."**
>
> **"Yeah yeah," said Zaphod. "Look, I think I'll just use a piece of paper."**
>
> **"Sure thing," said the computer. "I understand. If you ever want..."**
>
> **"Shut up!" said Zaphod, and snatching up a pencil sat down next to Trillian at the console.**
>
> **"OK, OK ..." said the computer in a hurt tone of voice.**
>
> *Douglas Adams (1986), The Hitchhiker's Guide to the Galaxy, London: Guild Publishing.*

The nature of interactions with the real world clarifies the notion of planning, which is a long-established AI research area. In everyday usage, the word 'planning' (in, for example, planning a holiday, planning how to carry out a complicated surgical operation, and so on) refers to the determination of some activities that are to be carried out at a later time. The planning process for, for example, a surgical operation, may involve the computational or mental simulation of possible sequences of actions: these sequences could not be carried out in reality (in the order that they are investigated during planning) because they may well be irreversible. Only when the plan is complete is any action carried out. Many AI programs, however, do not actually carry out the actions they determine. For example, a chess program determines a move, without physically making it. Thus, the distinction between problem solving and planning is often blurred.

However, sometimes the distinction is crucial because, in particular, the success of the plan depends on the predictability of the situation. An AI program might attempt to predict the possible outcomes of intermediate actions and develop contingent plans (which would be very time-consuming

if there are many possibilities most of which are unlikely to occur), or it might develop a provisional plan, open to modification if the outcomes as the plan is executed are not as anticipated:

It is a bad plan that admits of no modification.

Publilus Syrus (1st century B.C.), Moral Sayings.

The best-laid schemes o' mice an' men
Gang aft agley,
An' lea'e us nought but grief an' pain,
For promis'd joy!

Robert Burns (1785), To a Mouse, on disturbing her nest with a plough.

These oft-quoted lines of Burns's poem are immediately followed by a lament that, unlike the mouse, who is concerned only with the present, man can look backward and forward on "prospects drear", which is a less optimistic view of "our unique human capacity to transcend the present" than that of Betty Freidan above.

If during the execution of a plan something unanticipated occurs, it may not be necessary to abandon the plan completely. An on-the-fly modification may be sufficient to overcome the problem but determining this may require subtle reasoning about the interactions between components of the plan, and hence it is important to retain a record of these from the initial planning stage. Many of the AI planning techniques were developed during the STRIPS project of the early 1970s. STRIPS was the planning component of the Shakey robot developed at SRI International. In STRIPS, actions were defined in terms of their preconditions and effects. Preconditions have to be true before the action can be carried out (for example, the robot might be able to pick up a block only if there is nothing on top of it) and the effects describe changes that occur as a result of the action (for example, after a block is picked up it is in the robot's hand and no longer on the floor). Planning involved a search through sequences of possible actions to find a state of the world matching the goal (such as, having all blocks in a corner of the room). However, the STRIPS language was limited in requiring precise descriptions of preconditions and effects, when often it is difficult to say exactly what these are (for example, the action of picking up a block may accidentally move another one).

For a time, more powerful formal languages for planning were experimented with but on the whole these were found intractable. In situations were they seem necessary, such as a rapidly changing world, it is often better to be more cautious in planning: to determine one step at a time perhaps and re-plan from there. Most practical planning systems today can be seen as elaborations and extensions of the original STRIPS approach.

Even STRIPS recognised that plans needed to be contingent and their execution carefully modified, not simply developed and then executed, because, as anyone who saw Shakey in action appreciated, its actions were not very reliable: an instruction to move forward three feet did not mean that it moved exactly three feet.

The apparently sharp distinction that some people thought early AI drew between planning and acting led, as such matters naturally do in AI, to a hostile controversy, with one side claiming that, on the contrary, there was no such thing as planning. Actions just occurred in response to a particular context:

> **On the planning view, plans are prerequisite to and prescribe action ... The alternative view ... is that while the course of action can always be projected or reconstructed in terms of prior intentions and typical situations, the prescriptive significance of intentions for situated action is vague.**
>
> *Lucy Suchman (1987), Plans and Situated Actions: the Problem of Human-Machine Communication, New York: Cambridge University Press.*
>
> **Rather than relying on reasoning to intervene between perception and action, we believe activity mostly derives from very simple sorts of machinery interacting with the immediate situation. This machinery exploits regularities in its interaction with the world to engage in complex, apparently planful activity without requiring explicit models of the world.**
>
> *Philip Agre and David Chapman (1987), Pengi: an implementation of a theory of activity, Proceedings of AAAI-87, 268-272.*

In other words, while we may be able to interpret our own or somebody else's actions in terms of plans, there probably aren't any, and, as far as systems are concerned, they may not need plans or models of the world. Because we cannot anticipate eventualities, we react to the situation we are in. Actually, the thesis of Lucy Suchman, an anthropologist, is subtler than that of reactive AI. Her view is that action is improvised within a social context and that plans are just one resource in deciding what to do, rather than a mechanism to control action. Obviously, the emphasis on the social context is a contrast to the individualised view of problem solving that has been considered so far and, if valid (which it surely is in many cases), suggests that AI needs to take account of sociology as well as psychology. This view of 'situated action' became broadened into 'situated cognition' that, among other things, argued that the symbolic representation approach to AI was misguided.

The implications of such studies of human planning and acting for system design are not altogether clear. If the system is to be used in a social context, as sociologists would emphasise, with humans present and able to improvise to overcome any deficiencies then perhaps systems do not need to

be so carefully designed. If it is the case that humans do not just carry out plans then this does not necessarily imply that systems should not. Humans, after all, have limitations, for example, in memory, which systems may not. As far as the planning controversy goes, when the dust had settled, it became clear that in some cases it is necessary and possible to determine a sequence of actions in advance of when they are carried out, in other cases it is not possible or sensible to do so, and most cases fall somewhere in between – as is recognised in current AI planning schemes.

Undaunted by the thought that plans may not even exist or may not be needed, planning researchers have persevered and indeed have claimed major breakthroughs:

> **The current level of performance is quite impressive, with several planners quickly solving problems that are orders of magnitude harder than the test pieces of only two years ago.**
>
> *Daniel Weld (1999), Recent advances in AI planning, AI Magazine, 20, 2, 93-122.*

This is achieved mainly by the use of new techniques (embodied first in a system called Graphplan developed by Avrim Blum and Merrick Furst in 1995) to convert a planning problem into that of proving that a proposition is true and deriving a plan from the proof constructed, a problem that we will consider shortly. The incorporation of various other AI techniques to improve efficiency ensures that there continues to be a sufficient engagement in the classical planning problem (that of mapping from an initial state into a final state via a set of actions, assuming that time is discrete, that there are no external events, that the system knows everything that is relevant and that the effects of all actions are known) to enable researchers to rather disregard the view that people react, rather than plan, and so should systems. Consequently, planning systems raise the question, common in AI, of where the credit for success (if it is achieved) should be attributed:

> **AI planning theory has had a number of practical applications, and is one of the success stories of AI. However, practical applications of AI planning theory have been largely confined to well behaved domains in which goals are fixed and all the relevant information can be precompiled and supplied to the planner … the human must prepare the ground very carefully, being sure to give the planner all the knowledge needed to solve the planning problem.**
>
> *John Pollock (1998), The logical foundations of goal-regression planning in autonomous agents, Artificial Intelligence, 106, 267-334.*

16. Intelligence: "the wellspring of life"

Probably the first planning program, the Logic Theorist (GPS's predecessor), had been the only operating program (as opposed to theory, speculation and proposal) presented at the first meeting on Artificial Intelligence, a term believed to have been coined by John McCarthy while preparing the case for support for the meeting:

> **We propose that a two-month, ten-man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.**
>
> *John McCarthy, Marvin Minsky, Nathaniel Rochester and Claude Shannon (1955), proposal to the Rockefeller Foundation for the Dartmouth Summer Research Project on Artificial Intelligence.*

We have already met McCarthy and, briefly, Minsky and Shannon. McCarthy, the inventor of Lisp, and Minsky established the AI Laboratory at the Massachusetts Institute of Technology in 1958 and it was the pre-eminent research centre for the first decade or two of AI. McCarthy moved on to Stanford University in 1963 and Minsky to the MIT Media Laboratory in 1973. Shannon and Rochester were already well known for their work on information theory and the design of the IBM 701 (the first profitable computer), respectively, but neither did much work directly on AI. In addition to Newell and Simon, the other four participants were Trenchard More (of Princeton), Arthur Samuel (IBM), Oliver Selfridge (MIT) and Ray Solomonoff (MIT), the last three of whom went on to make significant contributions to AI.

Notwithstanding the illustriousness of the participants, the meeting was generally considered a disappointment. In particular, the Logic Theorist had a lukewarm reception, despite the fact that it initiated several themes which came to characterise AI research, such as: the use of working programs to refine psychological theories; the focus on problems which would previously have been conceded to require intelligence in their solution; the use of symbolic programming within specialised programming languages designed for AI problems; and the explicit mimicry of human intellectual processes as elaborated from detailed studies of human problem solving transcripts.

It is often said that the main outcome of the meeting was an agreement to adopt the term 'artificial intelligence' (already embedded in the workshop

title), which does not seem much of a return for such a concentrated effort. On first acquaintance, the term is unsettling at best and antagonistic at worst, which is not helpful for a research field dependent on public understanding. Confusion arises partly because the adjective 'artificial' may be used in two ways. With a phrase like 'red pen' we infer that an object is both red and a pen. Similarly, with 'artificial insemination', something is both artificial and insemination. It is considered artificial not because the end product is different but because the process of attaining it is unnatural. However, with 'artificial diamond' not only is the process unnatural but also the end product is certainly not a diamond. With most phrases of the form 'artificial X', such as 'artificial laugh', we feel that both the process and the product are not entirely genuine. As we will see, criticisms of AI are sometimes directed at the intelligence exhibited and sometimes at the process through which it is attained.

So an artificial intelligence is not necessarily an intelligence, whatever that may be, although some, at least, would not have objected to the choice of term:

The first duty in life is to be as artificial as possible. What the second duty is no one has yet discovered.

Oscar Wilde (1891), Phrases and Philosophies for the Use of the Young, in The Chameleon, Oxford.

The sad thing about artificial intelligence is that it lacks artifice and therefore intelligence.

Jean Baudrillard (1987), Cool Memories.

As might be predicted, contemporary post-modernists, of whom Baudrillard is the high priest, try to unsettle us by suggesting that artifice is desirable rather than reprehensible.

Intelligent people naturally assume that intelligence is an unquestionably positive attribute: they tend to ask if there is intelligent life elsewhere in the universe as though any other kind would be of little account. But intelligence is not universally acclaimed:

Long live death! Down with intelligence!

General Millán Astray (1936), slogan in the Spanish Civil War.

Intelligence .. nothing has caused the human race so much trouble as intelligence.

Stella, in the film Rear Window (1954).

So far as I can remember, there is not one word in the Gospels in praise of intelligence.

Bertrand Russell (1932), Education and the Social Order, London: Allen and Unwin.

Russell was a sufficiently committed atheist to have studied the Bible deeply in order to be sure of what he could not accept and he must therefore have forgotten the Book of Proverbs, which extolled the benefits of intelligence:

Intelligence is the wellspring of life unto him that hath it.

Bible, Proverbs 16, 22.

At least, the word 'intelligence' is one translation of the original Hebrew. Other translations prefer the word 'understanding' or 'wisdom'. Of course, we cannot now be sure of the shades of meaning intended by the original words.

For what it is worth, the word 'intelligence' is derived from the Latin 'inter' and 'legere' and originally meant 'an ability to choose between'. What it has evolved to mean today is unclear:

Intelligence is that faculty of mind, by which order is perceived in a situation previously considered disordered.

Haneef Fatmi and R.W. Young (1970), A definition of intelligence, Nature, 228, 97.

Intelligence is quickness in seeing things as they are.

George Santayana (1920), Little Essays, drawn from the writings of George Santayana, edited by Logan Pearsall Smith, New York: Ayer Company.

Intelligence is silence, truth being invisible.

Ned Rorem (1967), Random Notes from a Diary, Music from Inside Out.

… a person's intelligence is directly reflected by the number of conflicting points of view he can entertain simultaneously on the same topic.

Lisa Alther (1976), Kinflicks, New York: Knopf.

It is commonly thought to be a great scandal that psychologists cannot agree on a definition of intelligence ... any sentence starting 'intelligence is ...' justifiably arouses one's suspicions.

T.R. Miles (1957), On defining intelligence, British Journal of Educational Psychology, 27, 153-167.

To avoid the suspicions aroused by sentences starting with 'Intelligence is …', Randall Davis, then president of the American Association for Artificial Intelligence, tried to regard the word 'intelligence' as a collective noun, that is, a word singular in form but used in the plural:

Intelligence are many things … human intelligence is a natural artifact, the result of the process of evolution and its parallel, opportunistic exploration of niches in the design space. As a result, it is likely to bear all the hallmarks of any product of that process – it is likely to be layered, multifaceted, burdened with vestigial components, and rather messy.

Randall Davis (1996), What are intelligence and why?, AI Magazine, 19, 1, 91-110.

But, as we can see, he was unable to sustain this for long.

The multifaceted nature of intelligence seems a sensible thing to emphasise, given the difficulty of providing a precise definition:

> **Intelligence seems to denote little more than the complex of performances which we happen to respect but do not understand.**
>
> *Marvin Minsky (1963), Steps towards artificial intelligence, in Edward Feigenbaum and Jerome Feldman (eds.), Computers and Thought, New York: McGraw-Hill.*

It is a favourite gambit in AI to 'define' difficult concepts such as intelligence, creativity, emotion, consciousness, and so on, as just the set of things we don't understand. However, definitions such as Minsky's will not suffice even as an aphorism: it is easy to think of performances that we might respect but do not understand (such as, cartoonists producing instant caricatures, ballet dancers performing on pointed toes, memory men recalling baseball scores, and so on) and which we would not necessarily consider a manifestation of intelligence. And what does it mean to understand a performance and why should it be disqualified from intelligence if we do?

Another (apparently contradictory) gambit is to assert that any particular performance is really just a form of some other performance – so problem solving is a form of reasoning, natural language generation is a form of planning, learning is a form of problem solving, and so on. Hence there is really only one performance, that corresponding to intelligence. Nonetheless, it is possible to take Minsky's definition and begin by accumulating a list of abilities that we would expect an intelligent agent (human, computer, or whatever) to possess. Any agent that possessed a quorum of those abilities might be deemed intelligent. Unfortunately, such a list is not static. As we begin to discover how an ability (for example, to solve crosswords) is performed we tend to be less impressed by that ability. As we saw, arithmetical abilities were downgraded by the building of calculating machines. With computers, this belittling of previously intelligent abilities has accelerated:

> **Once some mental function is programmed, people soon cease to consider it as an essential ingredient of 'real thinking'. The ineluctable core of intelligence is always in that next thing which hasn't yet been programmed.**
>
> *Douglas Hofstadter (1979), Gödel, Escher, Bach: an Eternal Golden Braid, New York: Basic Books.*

The unfortunate consequence of the view that once something has been successfully programmed then it can't be AI is that AI is necessarily concerned with unsuccessful programs.

17. The Turing test: "a fundamental misunderstanding"

How then are we to decide if a computer can be justly described as intelligent? Again, Alan Turing had considered this question before the term 'artificial intelligence' existed. His answer, basically, was to pass the buck: if we are unable, on the basis of what it does, to distinguish between a machine and a human (presumed for the sake of the argument to be intelligent) then the machine may be considered intelligent, or at least as intelligent as the human. The version of this test that he described has come to be known as 'the Turing test'. An inquisitor is required to grill both machine and human to see whether one set of responses is distinguishably humanoid. Here is a snippet of a conversation that Turing suggested an intelligent machine (or human) should be able to take part in:

"In the first line of your sonnet which reads, "Shall I compare thee to a summer's day?" would not 'a spring day' do as well or better."

"It wouldn't scan."

"How about, 'a winter's day'? That would scan all right."

"Yes, but nobody wants to be compared to a winter's day."

"Would you say Mr. Pickwick reminded you of Christmas?"

"In a way."

"Yet Christmas is a winter's day, and I do not think Mr. Pickwick would mind the comparison."

"I don't think you're serious. By 'a winter's day' one means a typical winter's day, rather than a special one like Christmas."

Alan Turing (1950), Computing machinery and intelligence, Mind, 59, 236, 433-460.

This human-computer comparison is today taken to be the Turing test but is in fact a simplified form of that originally proposed. Turing developed his version of the test via an 'imitation game' in which an interrogator had to determine which of two people, a woman (obliged to tell the truth) and a man (obliged to lie), is the woman – a game with some poignancy in view of Turing's later arrest and eventual suicide because of his homosexuality.

Even though the Turing test is now enshrined in an annual competition, with a prize of $100,000 for the programmer of any system that passes the test, it needs to be said that, while it provides a convenient starting point in the search for AI, by no means everyone agrees that it is appropriate for the field. The test was originally framed to answer the question "Can machines think?" and it therefore equated intelligence with thinking and assumed that thinking is made overt by language, both questionable assumptions. It also

assumed a particular view of the computer, understandably derived from the Turing machine, as enabling interaction with a single user. Others, especially as computer technology has evolved, have a different view:

> **The futility of the Turing test comes … from a fundamental misunderstanding of the nature of computers and society as closed, centralized, and asocial. As that misunderstanding gets replaced by an open system, ecological, and political model of organizations, workplaces, and situations (which include both machines and human organization), the Turing test will be replaced by different forms of evaluation.**
>
> *Susan Leigh Star (1989), The structure of ill-structured solutions: boundary objects and heterogeneous distributed problem solving, in Les Gasser and Michael Huhns (eds.), Distributed Artificial Intelligence, Volume II, San Mateo, California: Morgan Kaufmann.*

Star then proposed instead a 'Durkheim test', in honour of the French sociologist Emile Durkheim (1858-1917), who argued that 'social facts' have to be understood at the system level and did not reduce to properties at the individual level.

The proposed Durkheim test has been ignored and the Turing test continues as AI's holy grail. One textbook even uses it in a definition of AI, with a clause that the test must be passed by a physical symbol system:

> **Artificial intelligence is the enterprise of constructing a physical-symbol system that can reliably pass the Turing test.**
>
> *Matt Ginsberg (1993), Essentials of Artificial Intelligence, San Mateo, Calif.: Morgan Kaufmann.*

The perceived importance of the physical symbol system hypothesis is also indicated by Mark Stefik's 870-page *Introduction to Knowledge Systems* (1995), which quotes it in the first sentence of chapter 1. The intent of adding the condition that the system must be a physical symbol one is to exclude from AI the possible passing of the test by 'inappropriate' means. For example, we might imagine that future computer memories will be so immense that all possible conversations could be stored verbatim and the system just has to retrieve the required response. (Computer chess programs rely on large databases of openings and endgames – imagine these somehow extended to cover the whole game, and then imagine doing the same for conversations). More pertinently, the definition excludes from AI the growing areas of research that do not accept the physical symbol system hypothesis as the only possibility in the design of intelligent machines.

It is natural for sciences to assert their legitimacy by defining boundaries to exclude fringe ideas – for example, medical science consigns some practices to 'alternative medicine' or even quackery – but it is surely

premature to exclude from AI those who do not accept one particular hypothesis, even though it is one that is so vaguely worded that it might not prove that exclusive:

> **If we define 'symbol' narrowly, so that the basic components in connectionist systems or robots of the sort advocated by Brooks are not regarded as symbols, then the [physical symbol system] hypothesis is clearly wrong, for systems of these sorts exhibit intelligence. If we define symbols (as I have, above) as patterns that denote, then connectionist system and Brooks' robots qualify as physical symbol systems.**
>
> *Herbert Simon (1995), Artificial intelligence: an empirical science, Artificial Intelligence, 77, 95-127.*

So, one of the proponents of the physical symbol system hypothesis can generously stretch the notion of a 'symbol' to encompass reactive robots, even though Brooks considers them to be 'without representation', and connectionist systems (to be discussed later), even though their designers consider them to be 'sub-symbolic', both lines of research in which proponents see themselves as engaged in an insurrection against the prevailing orthodoxy.

The Turing test is reminiscent of an interaction with an oracle, which was probably the very first manifestation of the idea of a 'thinking machine'. Oracles were devised by Egyptians in about 2500 B.C. Citizens went for advice to oracles, which were statues within which priests were hidden. Responses were conveyed by women sitting on tripods nearby. If the responses were sufficiently wise and insightful, then presumably citizens would trust in the oracle's omniscience. The skill of the oracle, like that of the astrologer today, lay in providing advice of such equivocality that the recipient would find insight in it, whatever happened.

Today it is not clear whether Turing intended his proposed test as a thought experiment serving as a basis for philosophical discussion or as an operational test to circumvent such discussion. Its adoption for the latter purpose continues to be controversial. At all events, despite arguments about its irrelevance or obsolescence, it seems likely that the Turing test will be with us for a while yet:

> **Despite all the controversy surrounding it, no other test has offered a viable alternative to the Turing test and it can expected to guide the field of AI for several more decades:**
>
> *Appa Rao Korukonda (2003), Taking stock of Turing test: a review, analysis and appraisal of issues surrounding thinking machines, International Journal of Human-Computer Studies, 58, 240-257.*

18. Language: "the index of his understanding"

The Turing test focuses on the use of language as a defining feature of intelligence. It assumes that language is an external manifestation of the internal capability that we wish to assess:

Language use epitomizes intelligent behavior.

Fernando Pereira and Barbara Grosz (1993), Introduction, Special Volume on Natural Language Processing, Artificial Intelligence, 63, 1-16.

He gave man speech, and speech created thought.

Percy Bysshe Shelley (1820), Prometheus Unbound, II, iv 72.

Clearly, Shelley, like his wife, the author of *Frankenstein*, was fascinated by the Greek legend of Prometheus, who made men of clay and animated them with fire from heaven. Unlike *Frankenstein*, however, his long poem *Prometheus Unbound* expresses optimism for the future of mankind.

The priority of the various human faculties that might be taken as a measure of intelligence remains a matter of debate, as we will see. On the face of it, though, it seems unlikely that speech necessarily precedes thought or that speech is a necessary measure of thought. In any case, until speech recognition and generation systems are much improved, the Turing test is, in practice, based upon communication via written or typed language, which is a very different matter.

Language began as a spoken form about 150,000 years ago, judging from the fact that the parts of the brain (Broca's and Wernicke's areas) used for speech are as developed in *Homo neanderthalensis* as they are in modern humans and much larger than they are in modern great ape brains. The oldest written records we have, from Sumeria, date from about 5,000 years ago. Written language, however, did not occur until after an intermediate stage in which there was communication via graphic images, which were the first attempts to overcome the ephemeral nature of speech. The earliest known cave paintings are about 40,000 years old. Since other animals (such as monkeys, birds and dolphins) communicate via sound, that is, a spoken language, a case can be made that it is the advent of graphical communication, not text-based language, which marks a defining achievement of human intelligence.

If that is so, then we should perhaps prefer a Turing test based on graphical communication and consider those AI programs concerned with vision. However, graphics (drawings, photographs, etc.) lack an important characteristic for intelligent reasoning: that is, they are inherently specific and cannot express general statements without some kind of abstract notation

formally equivalent to the kinds of symbolic logic considered shortly. It is possible to paint a buffalo chasing a man but it is not possible to paint something to indicate that all buffaloes will chase anybody:

The fact is that our pure sensuous concepts do not depend on images of objects, but on schemata. No image of a triangle in general could be adequate to its concept.

Immanuel Kant (1781), Critique of Pure Reason.

We might anticipate that a volume with the title of *Critique of Pure Reason*, written by an acknowledged great philosopher, Immanuel Kant (1724-1804), would be foundational for AI: we shall see.

The premise of the Turing test is that language-based communication provides a window into the communicator. The ability to sustain a conversation depends on the agent's knowledge of the rules and conventions of the language and of the topic of conversation and particularly, we would like to believe, since it is the basis for our multitude of written and oral examinations, on the agent's general intelligence and understanding:

A man's behaviour is the index of the man, and his discourse is the index of his understanding.

Ali Ibn-Abi-Talib (c650), Sentences, tr. Simon Oakley.

Like everything else this strange morning the words became symbols, wrote themselves all over the grey-green walls. If only she could put them together, she felt, write them out in some sentence, then she would have got at the truth of things.

Virginia Woolf (1927), To The Lighthouse, London: Hogarth Press.

There seems to be a belief that only through the use of language can understanding come about and that the use of language necessarily implies that such understanding exists.

Unfortunately, language can be a distorting window. We may use language poorly or deliberately to conceal intelligence and we may sometimes presume intelligence even if language indicates otherwise:

Where I am not understood it shall be concluded that something very useful and profound is coucht underneath.

Jonathan Swift (1704), A Tale of a Tub.

In *Gulliver's Travels*, Swift later considered the automatic generation of natural language. A professor of Laputa had invented a machine with wires linking hundreds of small cubes on the faces of which were Laputan words. Turning a crank produced sequences of words that, if meaningful, were copied down and built up into erudite treatises:

> **The most ignorant person at a reasonable charge, and with a little bodily labour, may write books in philosophy, poetry, politics, law, mathematics, and theology, without the least assistance from genius or study.**
>
> *Jonathan Swift (1726), Travels into Several Remote Nations of the World 'By Lemuel Gulliver'.*

At all events, science fiction writers have not hesitated to endow their robots with the ability to use language as we do. For example, Stanislaw Lem discussed how robots might come to acquire this ability:

> **Trurl put in six cliché filters, but they snapped like matches; he had to make them out of pure corundum steel. This seemed to work, so he jacked the semanticity up all the way, ... tossing out all the logic circuits, he replaced then with self-regulating egocentripetal narcissistors. The machine simpered a little, whimpered a little, laughed bitterly ... Then it asked for pen and paper. The next morning he went to see Klaupicius. Klaupicius, hearing that he was invited to attend the debut of Trurl's electronic bard, dropped everything and followed – so eager was he to be an eyewitness to his friend's humiliation ... Now the potentiometers indicated the machine's lyrical capacitance was charged to maximum, and Trurl, so nervous his hands were shaking, threw the master switch. A voice, slightly husky but remarkably vibrant and bewitching, said: "Phlogisticosh. Rhomothriglyph. Floof."**
>
> *Stanislaw Lem (1974), The Cyberiad: Fables for the Cybernetic Age, New York: Seabury Press.*

Trurl's electronic bard overcame this unpromising start to write poetry of such quality that human poets were driven to suicide, leading to its banishment to another planet. Actually, Stansilaw Lem was more a satirist of conventional science fiction than a science fiction writer himself. His *Cyberiad* whimsically explored philosophical issues surrounding alien intelligences and his earlier novel *Solaris* was converted into one of the very few films so far that considers the possibility that robots may have an intelligence quite unlike, rather than directly modelled on, that of humans.

Even humans require several years of careful training to be able to generate well-formed sentences reliably. In fact, we take longer to learn to generate sentences than we do to understand them. Perhaps, if we are to begin to develop computers that can deal with language we should start with the language understanding problem. Surprisingly, the first computer linguists chose a problem that compounded the difficulties of the language generation and understanding problems with the complication of attempting the two problems in different languages, that is, they tried to use computers to translate between languages. Naturally, they failed.

19. Pattern matching: "danger lurks there"

The first attempts at machine translation were made in the 1950s before the subject of artificial intelligence had been christened. The approach relied on vocabulary lookup, simple grammars, and statistics to resolve ambiguities, inspired by the success of computer-based statistical methods in cryptography to decode secret messages during the war:

> **It is very tempting to say that a book written in Chinese is simply a book written in English which was coded into the 'Chinese code'. If we have useful methods for solving almost any cryptography problem, may it not be that with proper interpretation we already have useful methods for translation?**
>
> *Warren Weaver (1949), Translation, in W.N. Locke and A.D. Booth, eds. (1955), Machine Translation of Languages, New York: John Wiley & Sons.*

For example, the translation of "The plane will not arrive" into "L'avion n'arrivera pas" depends on transcribing "plane" to "avion", on converting "le avion" to "l'avion", and on the system knowing that "plane" as aeroplane is more often associated with "arrive" than is "plane" as smoothing tool. As such, the method was fully in accord with the contemporary information theory and no doubt it was politically opportune to believe that it could be made to work.

But it could not and the coup de graçe was delivered in 1960 by Yehoshua Bar-Hillel who argued that the meaning of a sentence such as "The box is in the pen" could not be determined by any statistical association between "pen" and "box" but only from a commonsense understanding of the relative sizes and functions of various boxes and pens. So, in general, an automatic translator would need to possess an inconceivable volume of such commonsense knowledge of the world:

> **What such a suggestion amounts to, if taken seriously, is the requirement that a translation machine should not only be supplied with a dictionary but also with a universal encyclopedia. This is surely utterly chimerical and hardly deserves any further discussion.**
>
> *Yehoshua Bar-Hillel (1960), The present status of automatic translation of languages, in Franz Alt (ed.), Advances in Computers, Vol. 1, New York: Academic Press.*

At the same time, other difficulties with translation were being recognised:

> **Perhaps we could have a translation, I could not quite follow.**
>
> *Harold Macmillan (1960), responding to Nikita Khruschev banging his shoe on the desk at the United Nations on September 29.*

As a result, machine translation did indeed receive little further discussion for the two decades, except as a salutary illustration of the naive optimism of early AI research, even though no AI techniques were in fact used.

Today, however, machine translation is again an active area of research and development, tempered by realism, because, after all, even partly correct translations can be useful, as the translators provided with web browsers indicate. Even Bar-Hillel's chimera no longer seems unslayable. We can certainly provide a translation machine with access to an on-line encyclopedia. The difficulty lies, of course, in enabling the machine to make sense of it. In 1983 Douglas Lenat and colleagues began a ten-year project (called CYC) to achieve this, that is, to:

> **… represent a comprehensive corpus of real world knowledge (both the size and scope of the Encyclopedia Britannica) in a knowledge base, i.e. as a structured network of concepts, rather than a piece of text.**
>
> *Douglas Lenat, Alan Borning, David McDonald, Craig Taylor, and Steven Weyer (1983), Knoesphere: building expert systems with encyclopedic knowledge, Proceedings of the 8th International Joint Conference on Artificial Intelligence, 167-169.*

As CYC involves much more than language understanding, we will return to it later.

In 1960 nobody felt ready to tackle the problem of commonsense knowledge and the machine translation programme was shelved. Computational linguists retreated instead into formality, encouraged by two contemporary developments. First, Noam Chomsky's *Syntactic Structures*, published in 1957, had revolutionised linguistics and had provided a formal definition of what is now called the Chomsky classification of grammars into four types, a classification which is still the starting point for most work on formal grammars. Secondly, designers of programming languages had found it helpful to define their languages using a notation (Backus-Naur form) that is equivalent to one of Chomsky's four types of language, the context-free languages. It was recognised that grammars for programming languages were inadequate for natural languages:

> **The grammars that are used to describe computer languages are very much simpler than those that are needed for natural languages. Indeed the latter are by no means understood, and it is widely agreed that the kind of grammar which we shall be using is not satisfactory for natural languages.**
>
> *J.M. Foster (1970), Automatic Syntactic Analysis, Amsterdam: Elsevier.*

Nonetheless, the next phase of research began by hoping that computers could begin to get a grip on natural languages in ways similar to how

programming languages are handled, that is, by having a formal grammar to define the syntax of the language and by compiling the syntactic structures as generated by the grammar into some 'object code' representing the sentence's meaning.

However, the grammar-based approach to language understanding tended to ignore the distinction between 'competence' and 'performance' that Chomsky had emphasised. His formal grammars focussed on competence, that is, with the implicit knowledge of an ideal speaker-hearer of the language, whereas any human-computer interaction is necessarily concerned with performance, that is, the actual use of the language. Consequently, the formal grammarian's development of detailed grammars that were capable of generating scores of alternative structures for sentences in which native speakers found no ambiguity at all eventually led to the view that the definition of a grammar for a natural language was not the major problem:

"I don't want to talk grammar. I want to talk like a lady."

Eliza, in George Bernard Shaw (1916), Pygmalion.

The original Pygmalion, from Ovid's *Metamorphoses*, was a king of Cyprus who, finding fault with all women, created a statue of the perfect woman and married it (or her) after Aphrodite brought it to life. In *Pygmalion*, Eliza is trained in English speech and the class system and, in a familiar plot, rebels against her mentor to achieve freedom and independence.

The Eliza approach to language inspired Joseph Weizenbaum to develop his now notorious program ELIZA to investigate how natural language communication may be carried out without any detailed analysis of the sentences. Here is a short extract from a conversation with ELIZA:

"You are not very aggressive but I think that you do not want me to notice that."

"What makes you think that I am not very aggressive?"

"You don't argue with me."

"Why do you think I do not argue with you?"

"You are afraid of me."

"Does it please you to believe I am afraid of you?"

Joseph Weizenbaum (1967), Contextual understanding by computers, Communications of the ACM, 10, 474-480.

Is it obvious which of the participants in this dialogue is ELIZA and which the human? The core of the ELIZA program is a long list of expected phrases, such as "I hate X" where X stands for any sequence of words, such as "communists" or "being confused". Associated with each such phrase is one or more possible responses, such as "Why do you hate X?" or "Have you always hated X?" Whenever the input sentence matches one of the expected

phrases, ELIZA outputs one of the responses. If there is no match, ELIZA selects a general alternative, such as "Tell me about your family", or returns to a previous topic. So ELIZA's contribution to the conversation does not derive from any understanding, in any significant sense of the word, of the input sentences:

> **They had no conversation properly speaking. They made use of the spoken word in much the same way as the guard of a train makes use of his flags, or of his lantern.**
>
> *Samuel Beckett (1951), Malone Dies (English translation 1956).*

Samuel Beckett had the novel strategy, for an Irishman, of writing in French, because he welcomed the constraints imposed by writing in a non-native language, leaving others to translate into English. Possibly, this has implications for the machine translation enterprise.

To dismiss ELIZA as a trick, as many do, is to miss the point. In what sense is ELIZA any more of a trick than any computer program would have to be to communicate in natural language? The answer is that Weizenbaum had carefully contrived a situation in which the inadequacies of the phrase-matching technique were to a large extent hidden. First, the users of ELIZA were beguiled into attributing more intelligence to ELIZA than it actually possessed – once a user suspects that ELIZA really understands nothing of what is being discussed it is easy to find input sentences to demonstrate this. But even today the winners of the annual Turing test competition use ELIZA-like techniques. Secondly, ELIZA's role of psychotherapist was chosen precisely because it is the only role in which one is expected to ask (possibly foolish) questions and to never contribute anything new to the conversation.

ELIZA shows that very simple techniques can be effective in certain contexts. An even earlier natural language understanding program had already demonstrated this. The program was supposed to know a joke that an interrogator had to ask yes/no questions to discover. People would spend hours with this program, typing in "Is it about women?", "Is it a blue joke?", and on and on. However, such pattern-matching techniques cannot be extended to provide a general natural language communicating capability, as Weizenbaum was well aware:

> **ELIZA shows, if nothing else, how easy it is to create and maintain the illusion of understanding, hence perhaps of judgement deserving of credulity. A certain danger lurks there.**
>
> *Joseph Weizenbaum (1966), ELIZA: a computer program for the study of natural language communication between man and machines, Communications of the ACM, 9, 36-45.*

The danger did not lurk for long. Within a few years, as we will see, it was out in the open and so alarming Weizenbaum that he came to consider AI research as unethical because it would lead to a dehumanisation of humanity. (Incidentally, the 'joke program' simply answered "yes" if the question ended with a 's' or 'n', "no" otherwise.)

20. Procedural semantics: "a superficial and misleading way"

As far as natural language understanding was concerned, the lack of progress during the 1960s, compared to initial expectations, from both the formal grammar and ad-hoc approaches contributed to a general feeling of despondency in the AI field. Outsiders began to be deeply critical of the whole endeavour. For example, Sir James Lighthill, a distinguished applied mathematician, wrote a damning review of AI research, commissioned by the UK Research Councils and published in 1972:

> **AI not only fails to take the first fence but ignores the rest of the steeplechase altogether.**
>
> *James Lighthill (1972), Artificial Intelligence: A General Survey, in Artificial Intelligence: a paper symposium, London: Science Research Council.*

At that time, Lighthill held the Lucasian chair of Mathematics at Cambridge (once held by Newton) and in his illustrious career the AI review was but a bold diversion – as indeed was his sadly unsuccessful attempt in 1998 to swim the ten miles around the island of Sark at the age of 74.

A last-minute postscript had to be added to his review specifically to dismiss Terry Winograd's remarkable thesis, completed in 1971, as the swallow that doesn't make a summer. The apparent ability of his natural language program SHRDLU to communicate in English, as shown in the most famous example of human-computer interaction, was way in advance of anything seen earlier:

...

User: **Which cube is sitting on the table?**
SHRDLU: **The large green one which supports the red pyramid.**
User: **Is there a large block behind a pyramid?**
SHRDLU: **Yes, three of them: a large red one, a large green cube, and the blue one.**
User: **Put a small one onto the green cube which supports a pyramid.**
SHRDLU: **OK.**
User: **Put the littlest pyramid on top of it.**
SHRDLU: **OK.**

User: **Does the shortest thing the tallest pyramid's support supports support anything green?**

SHRDLU: **Yes, the green pyramid.**

...

Terry Winograd (1972), Understanding Natural Language, New York: Academic Press.

SHRDLU, by the way, was a sequence of letters on a typesetter's keyboard and was often used by proofreaders as a temporary and glaringly meaningless patch to be repaired later. Consequently, SHRDLU occasionally crept through to make mysterious appearances in the columns of newspapers. Shrdlu was also a character, a proofreader, in fact, in Elmer Rice's play *The Adding Machine* (1923), in which the main character, Mr Zero, after 25 years number crunching is made redundant by an adding machine and in response murders his employer (well, it makes a change from machines murdering humans). Even so, Mr Zero makes it to heaven after execution and meets up with Shrdlu, who had found proofreading too much and in a moment of aberration had cut his mother's throat instead of the Sunday roast.

The behaviour of the program SHRDLU was equally inexplicable for a time. It was not at all clear which aspects of the program's design were responsible for such a startling improvement in capability. Winograd himself emphasised several points. First, SHRDLU did not have a rigid demarcation between syntax, semantics and reasoning, as the formal grammar approach proposed – but, in fact, SHRDLU's use of traditional parsings now seems fairly old-fashioned. Secondly, Winograd argued that general theorem-proving techniques (to be discussed shortly), which were being advocated for question answering, were too inefficient and needed to be supplemented by problem-specific knowledge – but, again, in retrospect, it seems that SHRDLU re-expresses, rather than discards, the logical approach. Finally, and perhaps most importantly, Winograd emphasised that language understanding depends partly on specific relevant world knowledge, which needs to be represented procedurally, that is, as pieces of program, which can be run when needed.

So, following Bar-Hillel's conclusions concerning the need for world knowledge in machine translation, Winograd aimed to give his program deep knowledge of its world, the so-called blocks world, in which a simulated robot arm could move blocks about on a tabletop:

You blocks, you stones, you worse than senseless things.

William Shakespeare, Julius Caesar, 1, 1.

Rather ironically, far from tackling the problem of world knowledge, the blocks world enables it to be ignored. None of the concerns of the real world arise, as John Haugeland demonstrated with his SHRDLU parody:

Trade you a squirtgun for a big red block.
Sorry, I don't know the word "trade".
A "trade" is a free exchange of property.
Sorry, I don't know the word "free".
A "free" act is done willingly, without being forced.
Sorry, I don't know the word "act".
"Acts" are what people do on purpose, and not by accident.
Sorry, I don't know the word "people".
Sorry, I thought you were smarter than you are.
Sorry, I don't know the word "sorry".

John Haugeland (1985), Artificial Intelligence: the Very Idea, Cambridge, Mass.: MIT Press.

This is not a problem of vocabulary: SHRDLU just does not possess the concepts needed to make sense of "trade", "free", or any other everyday concern. There could not be a 'trade world', comparable to the blocks world, because trade is not something that can be hermetically sealed off from the rest of everyday life.

Fundamentally, this difficulty derives from SHRDLU's 'procedural semantics', that is, the representation of the meaning of sentences as procedures that could be executed to generate answers (for questions) or to perform actions (for commands), or stored to answer later questions (for statements). The assumption is that a sentence is to be compiled into a program representing the meaning of that sentence, such that when the program is executed it will determine whether the sentence is true:

In order for an intelligent entity to know the meaning of such sentences it must be the case that it has stored somehow an effective set of criteria for deciding in a given possible world whether such a sentence is true or false.

William Woods (1975), What's in a link?, in Daniel Bobrow and Allan Collins (eds.), Representation and Understanding, New York: Academic Press.

This is a re-statement of the verification principle of logical positivism and earlier philosophies. Is it really the case that one cannot understand the meaning of sentences, for example, "Oswald shot Kennedy" and "Newell liked bananas", without having a means to determine if they are true or false? The blocks world 'works' because it is contrived precisely so that the verification principle holds.

So, what happened after the apparently stunning breakthrough of 1972? Winograd himself came to consider the SHRDLU project misguided and

turned to more fundamental studies on the nature of human-computer interfaces. He came to the conclusion that the symbol-processing methodology of AI was unsound:

> **My own work underwent a major change, as I moved away from the assumption that the way to make better and more useful computers (and interfaces) was to get them to be intelligent and use natural language. I recognized the depth of the difficulties in getting a machine to understand language in any but a superficial and misleading way, and am convinced that people will be much better served by machines that do well-defined and understandable things than those that appear to be like a person until something goes wrong (which won't take long), at which point there is only confusion.**
>
> *Terry Winograd (1985), in Daniel Bobrow and Patrick Hayes (eds.), Artificial intelligence – where are we?, Artificial Intelligence, 25, 375-415.*

It is not at all uncommon for AI researchers, after investing several years on a topic, presenting their achievements as positively as possible, and apparently reaching the verge of real success, to then turn away, possibly realising the enormity of the challenge still remaining and the likelihood of diminishing returns, to some new topic or to more basic research, sometimes explicitly disassociating themselves from the previous line of research. After all, if criticism is to come, as it surely will, who is in a better position to provide it?

However, in Winograd's case, the volte-face was not so startling as it appears. His focus on a procedural approach to semantics, arguing that a theory of language should focus on the cognitive processes of language use rather than on the linguistic objects produced, was already a robust challenge to the prevailing orthodoxy of linguistics at the time. In linguistics, the focus on competence rather than performance was intended to eliminate the theoretically uninteresting aspects of language, such as hesitations and slips, but had the effect of eliminating all factors concerned with cognitive processes. (It is a risky business to "eliminate the theoretically uninteresting aspects": Winograd did the same in having an imaginary world with no real robot moving real blocks. Others claim that language has to be grounded in the physical world.) The competence-performance distinction assumes that language can be formally described independently of its use. If this is denied, it leads naturally to a view that formal grammars and symbolic descriptions generally are not very useful and that a prior concern should be the social context of language use, which is soon found to be so complex that the AI ambition for natural language systems seems unrealistic.

Eugene Charniak, another pioneer of AI work on natural language, experienced similar disillusionment:

> **Few, if any, consider the traditional study of language from an artificial intelligence point of view a 'hot' area of research. A great deal of work is still done on specific natural language processing problems, from grammatical issues to stylistic considerations, but for me at least it is increasingly hard to believe that it will shed light on broader problems, since it has steadfastly refused to do so in the past.**
>
> *Eugene Charniak (1993), Statistical Language Learning, Cambridge, Mass.: MIT Press.*

In the 1970s he had tried to develop a system to understand children's stories such as "Today was Jack's birthday. Janet and Penny went to the store to get presents. 'I will get a top,' said Janet. 'Don't do that,' said Penny. 'Jack has a top. He will ask you to take it back.'" The difficulties he encountered, for example, in determining what the "it" refers to, had convinced some people (especially himself) of the impossibility of the AI language understanding programme. In his own case, he switched to the use of statistical methods for language processing, which are based on the insight that the ambiguity of a sentence such as "I like fascinating women" might be probabilistically resolved by considering the frequency with which pairs of words co-occur, without needing complex grammars.

Winograd's emphasis on procedural representations and their supposed advantages over 'declarative' representations was a contribution to a controversy that will always be present in AI. At least, it will be as long as AI is led by English-speakers. Other cultures may not have the confusions caused by using the word 'know' to mean so many different things. For example, the French 'savoir' and 'connâitre' capture something of the procedural (know how) – declarative (know that) distinction. The distinction has some psychological basis, as many experiments have shown that people can learn to do things without being able to verbalise what they have learned, and clearly people can declare what they 'know' without using that knowledge when it is needed.

The decision to represent the meaning of sentences as procedures relates to the general question of the appropriate form of internal representation to enable a system to maintain a natural language interaction. Other researchers sought to define the 'atoms' in terms of which the meaning of sentences could be expressed. For example, Roger Schank tried to define a small set of 'ACTs', such as MTRANS for the mental transfer of information and CONC for conceptualisation, to cover the meaning of all verbs:

> **... these fourteen ACTs plus a number of states will adequately represent the information underlying English verbs ... Since there are thousands of verbs and only fourteen ACTs for which inference rules need be written, this amounts to a tremendous saving and is probably quite a bit more like the way people operate.**
>
> *Roger Schank (1973), Identification of conceptualizations underlying natural language, in Roger Schank and Kenneth Colby (eds.), Computer Models of Thought and Language, San Francisco: W.H. Freeman.*

Schank showed how the meaning of a sentence such as "John hit Mary by throwing a stick at her" may be neatly, if not succinctly, represented by a 'conceptual dependency network' in which about twenty concepts are related by about the same number of ACTs, the idea being that the inference rules for the ACTs would help the system reason about following sentences, such as "Mary growled and bit John's leg".

On paper, these ACTs, networks and similar notations took on baroque forms that were comprehensible to only a few. Consequently, there was a move to replace all these ad-hoc notations by some standardised form that everyone could understand. In particular, the suggestion was to use an established notation, such as predicate logic (to be discussed shortly), with defined inference procedures. In other words, as far as natural language understanding was concerned, the idea was to parse sentences using grammars, to generate predicate logic expressions to represent the meaning of the sentences, and to use standard inference procedures to generate system responses. It may be a computational convenience to use standard inference procedures rather than have to write special ones to deal with even as few as fourteen basic concepts but the proposal is of dubious psychological and philosophical validity. It is unlikely that humans translate sentences into logic, if that matters, and the relationship between language and thought is controversial, to say the least.

21. Reasoning: "nothing but 'reckoning'"

Just as writers often point to the use of language as the defining difference between humans and animals, so philosophers often identify the ability to perform rational reasoning – we are, after all, *Homo sapiens*:

> **There are many gifts that are unique in man; but at the centre of them all, the root from which all knowledge grows, lies the ability to draw conclusions from what we see to what we do not see.**
>
> *Jacob Bronowski (1973), The Ascent of Man, Boston, Mass.: Little Brown & Co.*

The ability to draw conclusions has been a subject of study since antiquity – since Socrates drew up his tables of syllogisms, if not before – and the need to impose some rigour on the process was recognised long ago. Ramon Lull, a Spanish theologian, had a divine illumination in 1274 which led him to write his *Ars magna*, the first of about forty treatises that he wrote on a method using geometrical diagrams to discover non-mathematical truths. A summary of the method, which was the first attempt to use a mechanical device to operate as a logic system, concluded that it:

> **... amounted virtually to a satire of scholasticism, a sort of hilarious caricature of medieval argumentation.**
>
> *Martin Gardner (1958), Logic Machines and Diagrams, New York: Dover Publications.*

Lull had hoped to use the method to convince the heathen that Christianity is the one true faith.

Thomas Hobbes (1588-1679) also hoped to apply mechanical, rational processes to social issues, such as the avoidance of civil war. His ideas on politics, religion and the law were such that he had to escape from England from 1640 to 1651 and after his return he was forbidden to publish his opinions although he was allowed to publish a translation of the *Odyssey* at the age of 86. Inspired by Galileo's recent use of geometric methods to develop laws of motion, Hobbes hoped to apply similar methods to a geometrical deduction of the behaviour of people. He conceived of reasoning as the processing of symbols:

> **For 'reason' … is nothing but 'reckoning,' that is adding and subtracting, of the consequences of general names agreed upon for the 'marking' and 'signifying' of our thoughts … The use and end of reason is not the finding of the sum and truth of one or a few consequences remote from the first definitions and settled significations of names, but to begin at these, and proceed from one consequence to another. For there can be no certainty of the last conclusion without a certainty of all those affirmations and negations on which it was grounded and inferred.**
>
> *Thomas Hobbes (1651), Of Man, Being the First Part of Leviathan.*

This view of thinking as symbolic operations following systematic rules makes Hobbes yet another claimant to be the founder of AI.

Leibniz admired Hobbes's theories of the nature of thinking and was also inspired by Lull's work, bizarre as it now seems, towards speculating on the possibility of determining truth by some process of calculation:

> **I feel that controversies can never be finished ... unless we give up complicated reasonings in favour of simple calculations, words of vague and uncertain meaning in favour of fixed symbols ... when controversies**

> **arise, there will be no more necessity for disputation between two philosophers than between two accountants. Nothing will be needed but that they should take pen in hand, sit down with their calculators [counting tables], and ... say to one another: let us calculate.**
>
> *Gottfried Wilhelm Leibniz (1686), De Scientia Universali seu Calculo Philosophico.*

Leibniz had previously attempted to demonstrate the benefits of reducing reasoning to some form of calculation by 'proving', from about sixty propositions concerning the social and economic situation of Poland in 1668, that the German Count Philipp Wilhelm von Neuberg was the rational choice in an election of the next king of Poland. Unaccountably, the electorate were not persuaded by his logical argument and elected the Pole Michael Wisniowiecki instead. This demonstration of irrationality did not deter Leibniz from his vision of some kind of artificial language through which reasoning could be automated. He imagined a symbolism (a characteristica universalis) that could be manipulated "without any labour of the imagination or effort of the mind", to form a 'reasoning calculus', to go along with the differential and integral calculus that he had already invented.

It is apparent that these early visions of mechanised thought were enthused by dreams of the social benefits that would follow. Unfortunately, the visions were mirages. The means did not exist to bring them to reality. The suggestion of Leibniz that reasoning should be carried out with formal symbols and not with "words of vague and uncertain meaning" went largely unheeded for almost two centuries, with John Stuart Mill, for example, still insisting in 1843 that reasoning relied on the precise use of words:

> **Since reasoning, or inference, the principal subject of logic, is an operation which usually takes place by means of words, and in complicated cases can take place in no other way: those who have not a thorough insight into both the significance and purpose of words, will be under chances, amounting almost to certainty, of reasoning or inferring incorrectly.**
>
> *John Stuart Mill (1843), A System of Logic, Ratiocinative and Inductive, London: Longmans.*

Mill's *System of Logic* contained important comments on the nature of induction, causality and the philosophy of science but as far as logic and AI are concerned its main contribution was to clarify that meaning involved more than naming, with a distinction being made being 'singular names' (such as Gottfried, Wilhemina) and 'general names' or predicates as we would call them today (such as old, father, logician).

Even if reasoning was not accepted to be equivalent to symbol-processing, it was evident, at least to mathematicians, that symbols were an indispensable aid to reasoning:

> **The symbolic form of the work has been forced upon us by necessity: without its help we should have been unable to perform the requisite reasoning.**
>
> *Alfred North Whitehead and Bertrand Russell (1910), Principia Mathematica, Cambridge: Cambridge University Press.*

Within AI, the nature of computers and programming seemed to imply that some kind of formal language for reasoning was necessary to enable a computer to make decisions:

> **We want a computer program that decides what to do by inferring in a formal language that a certain strategy will achieve its assigned goal. This requires formalising concepts of causality, ability, and knowledge.**
>
> *John McCarthy and Patrick Hayes (1969), Some philosophical problems from the standpoint of artificial intelligence, in Bernard Meltzer and Donald Michie (eds.), Machine Intelligence 4, New York: American Elsevier.*

McCarthy and Hayes did not explicitly say what they intended by a 'formal language' but it was implicit that they had in mind a system of symbolic logic. They did not discuss, for example, whether something like Lull's geometrical diagrams would in principle count as a formal language, or indeed the many other diagrammatic or other forms of notation (such as musical scores, architectural plans, chemical structures, maps, and so on) which their users are able to reason with to come to decisions, without, it appears, any processes corresponding to a rigorous logical derivation.

They did, however, give examples using expressions written in modern symbolic logic, the foundations of which were laid by George Boole in 1847. He had had no reticence in proclaiming the significance of his *Laws of Thought*:

> **The laws we have to examine are the laws of one of the most important of our mental faculties. The mathematics we have to construct are the mathematics of the human intellect ... To unfold the secret laws and relations of those high faculties of thought by which all beyond merely perceptive knowledge of the world and of ourselves is attained or matured is an object which does not stand in need of commendation to a rational mind.**
>
> *George Boole (1854), An Investigation of the Laws of Thought.*

Although less well known, his earlier book *The Mathematical Analysis of Logic* (1847) was actually the first book of modern logic.

The laws that he developed define what is now known as propositional logic or, in a generalised form, Boolean algebra. Variables, `p`, `q`, `r`, etc. are used to denote propositions, that is, statements that are true or false (as opposed to expressions such as "Happy birthday", "Shakespeare's sister", and "Will you marry me?", which are not propositions). For example, `p` may denote the proposition that "Berlin is the capital of Germany". We may not know whether it is true or false but we may agree that it has to be one or the other:

It is more important that a proposition be interesting than that it be true.

Alfred North Whitehead (1933), Adventures of Ideas, Pt. III, Ch. 16, New York: New American.

In propositional logic connectives are defined to link propositions into more complex propositions. For example, `not`, `and`, `or` and `implies` are defined as follows:

`not p` is true only if `p` is false;

`p and q` is true only if both `p` and `q` are true;

`p or q` is true only if `p` is true or `q` is true or both are true;

`p implies q` is true only if `(not p) or q` is true.

In order to remove a confusing association with the English words "not", "and", "or" and "implies", arbitrary formal symbols such as: `~`, `&`, `v` and →, respectively, would be better, but we will continue to use the English symbols to aid (mis)readability, except for `implies`, which is particularly tricky.

Consider the statement "If I offended you then I am sorry". If we represented the meaning of this by a `p → q` formula we would mean that the statement is true provided that either I did not offend you (in which case it is irrelevant whether I am sorry) or I did offend you and I am in fact sorry. However, in propositional logic, there is no causal or any other relationship between the `p` and the `q` of `p → q`. For example, if `p` denoted "Toronto is in Mexico" and `q` denoted "kangaroos are insects" then `p → q` would be a true proposition. So we'll use the symbol → to emphasise its special character.

From these definitions it is possible to derive 'laws of thought' or conditions under which an argument that a conclusion follows from premises is sound. For example, if `p or q` is true and `p` is false, then it follows from the definition of `or` that `q` is true. A more complicated example is that if both the following are true

```
p or q
(not p) or r
```

then it follows that

```
q or r
```

is true. Instantiating (as they say) the variables, this might be an argument such as

"Today is Sunday or Saturday"

"Today is not Sunday or I am in church"

therefore

"Today is Saturday or I am in church".

From the definitions of the connectives, an alternative verbalisation of the above argument is:

"If today isn't Sunday then it's Saturday"

"If today is Sunday then I am in church"

therefore

"Today is Saturday or I am in church."

However, these verbalisations are potentially dangerous because they may lead us into a discussion of what exactly words like "if", "or", "then", and so on really mean. Propositional logic has nothing to say about this: it is a formal system defined in terms of arbitrary symbols.

22. Resolution: "sicklied o'er"

This particular 'rule of inference' may be expressed more formally as

$$\frac{(p \text{ or } q) \text{ and } ((\text{not } p) \text{ or } r)}{q \text{ or } r}$$

where the notation means that the expression below the line can be derived from that above the line by inference. Or, in formal symbols:

$$\frac{(p \vee q) \,\&\, (\sim p \vee r)}{q \vee r}$$

This rule is called 'resolution' and has become the cornerstone of computational logic since Alan Robinson showed in 1965 that an extended form of the rule is complete for the more powerful logic, predicate logic, which we will introduce shortly:

> **Thus the native hue of resolution is sicklied o'er with the pale cast of thought.**
>
> *William Shakespeare, Hamlet, 3, 1.*

By complete, we mean that any inference that should follow from a set of premises can be made using the rule of resolution alone. Most textbooks on logic still give long lists of inference rules despite the amazing fact (it seems amazing to me, anyway) that at most one of them (for they often don't include resolution in the list) is actually needed.

So, if we stick to propositional logic for the moment, we have the basis for an automatic method for drawing conclusions and hence for proving

theorems (that is, for showing whether or not a particular expression in the logic is true for any assignment of truth values to the variables in the expression): to draw a conclusion `c` we take the premises `p1`, `p2`, … `pn` and repeatedly apply the rule of resolution until the expression `c` is derived or the rule cannot be applied any more. Since resolution is complete and (for propositional logic) it can only be applied a finite number of times then this process is bound to end either with `c` derived or not. In the former case, we have shown that

```
p1 and p2 ... and pn → c
```

is a theorem.

A proof is a sequence of applications of the rules of inference of a logical system to the premises and expressions derived from them. Here, for example, is a propositional logic proof corresponding to an old argument that God does not exist (as already emphasised, the proof is really only concerned with the formal symbols on the left; the symbols' interpretations, indicated on the right, are up to us). There are six premises:

1.	`not p`	"God does not prevent evil"
2.	`(not a) → i`	"If God were not able to prevent evil then he would be impotent"
3.	`(not w) → m`	"If God were not willing to prevent evil then he would be malevolent"
4.	`(a and w) → p`	"If God were able and willing to prevent evil then he would do so"
5.	`g → (not i)`	"If God exists then he's not impotent"
6.	`g → (not m)`	"If God exists then he's not malevolent"

Before we can let the rule of resolution loose on these premises, they must be converted into a form (called conjunctive normal form) that can match the rule. This is a straightforward process, which in this case leads to the revised set of premises:

```
1'.  not p
2'.  a or i
3'.  w or m
4'.  (not a) or (not w) or p
5'.  (not g) or (not i)
6'.  (not g) or (not m)
```

Then five applications of the rule of resolution enable the desired conclusion to be drawn:

7.	`(not g) or w`	from 6' and 3'
8.	`(not g) or (not a) or p`	from 7 and 4'
9.	`(not g) or (not a)`	from 8 and 1'

10. `(not g) or i` from 9 and 2'
11. `(not g)` from 10 and 5'

that is, "God does not exist".

"Oh dear," says God, "I hadn't thought of that," and promptly vanishes in a puff of logic.

Douglas Adams (1986), The Hitchhiker's Guide to the Galaxy, London: Guild Publishing.

So it seems that quite intricate arguments might be amenable to formalisation and hence computerisation.

However, it is worth remarking at the outset that logic works by drawing a clear separation between the 'facts' (the premises or axioms, which are supposed to describe what is known) and the 'processes' (the rules of inference, which are supposed to describe how further facts may be derived from the known facts) – for this separation re-occurs in various guises (including some we have already met such as the generality of GPS and the declarative-procedural controversy). The rules of inference are supposed to be general, that is, to apply equally to any set of facts. The separation obviously means that it is possible to focus on either aspect without concern for the other aspect. For example, we might consider that facts exist and can be described regardless of how they may be used (similarly, perhaps, to linguists considering that formal grammars can be developed without considering language use). As a preliminary indication that the separation may not be clear-cut, consider how one might represent the fact that in a particular context one rule of inference is more useful than another.

The basic theorem-proving algorithm indicated above is 'mindless'. With most logical proofs we have to make decisions about which inference rules to apply and which expressions to apply them to but here there is only one rule and we apply it to all possible pairs of expressions:

I can stand brute force, but brute reason is quite unreasonable. There is something unfair about its use. It is hitting below the intellect.

Oscar Wilde (1891), The Picture of Dorian Grey, Philadelphia: James Sullivan.

The algorithm is also exceedingly tedious. As you have no way of knowing in advance which premises are relevant, you might have thousands of them. You have to try to apply the rule of resolution to every pair of premises, and then to every pair involving the results of applying the rule of resolution, and so on. In propositional logic this process is bound to end but not necessarily within a useful timescale.

In practice, rather than prove a theorem directly, it is usually easier to prove it by reductio ad absurdum, that is, by assuming the negation of the theorem and showing that this leads to a contradiction:

Utinam tam facile possem vera reperire, quam falsa convincere.
(Would that I could discover truth as easily as I can uncover falsehood – or … **as I can condemn falsehood**, according to some translators. Latin scholars, please advise**.)**

Cicero (44 B.C.), De Natura Deorum, I, 91.

The theorem-proving strategy, then, in both propositional logic and predicate logic, is to take the expression

```
not (p1 and p2 ... and pn → c)
```

which, from the definitions of the connectives, is equivalent to

```
p1 and p2 ... and pn and not c
```

and to apply resolution repeatedly until a contradiction is derived. A contradiction is indicated by the derivation of an expression of the form

```
q and not q
```

which the rule of resolution will reduce to nothing, that is, false.

The automatic proof of a theorem certainly seems impressive but it is important to be clear what has been achieved. We have not proved that the conclusion (`c`) is true. We have proved that the expression

```
p1 and p2 ... and pn → c
```

is true, provided we accept that the rules of inference used are sound. We have assumed the truth of the premises:

The conclusion of your syllogism, I said lightly, is fallacious, being based upon licensed premises.

Flann O'Brien (1939), At Swim-Two-Birds.

If any one of the premises happens to be false then we can prove any conclusion at all (because if `p` is false, then `p → q` is true for any `q`), so that case is of little interest. So our tendency to imbue 'logical conclusions' with an aura of necessary, incontrovertible truth should be tempered by the realisation that logic does not (in fact, cannot) say anything about the truth of the premises used.

The goal of 'truth' is certainly admirable:

Let us begin by committing ourselves to the truth, to see it like it is and to tell it like it is, to find the truth, to speak the truth and live with the truth. That's what we'll do.

Richard Nixon (8 August 1968), Presidential nomination acceptance speech.

Logic, however, is not concerned directly with the truth of expressions. It is concerned only with the soundness of arguments. Logic aims to provide a framework upon which a structure of reasoning may be built:

Logic is simply the architecture of human reason.

Evelyn Waugh, in Mark Amory, ed. (1980), The Letters of Evelyn Waugh, New York: Weidenfeld and Nicolson.

Of course, if the premises are in fact considered to be true and the rules defining the processes of reasoning are considered to be sensible, then we may reasonably expect to consider the conclusion to be true, as that is the whole purpose of the logical notation.

Although, rather ironically, Boole's propositional logic had been re-discovered by Claude Shannon in his 1937 thesis as an appropriate notation for describing the behaviour of switching circuits and hence for the 'logical design' of the basic hardware of computers, it was never really adopted to define laws of human thought, as Boole intended, despite the advocacy of (amongst others) the renowned developmental psychologist, but amateur logician, Jean Piaget:

> **No further operations need to be introduced since these operations correspond to the calculus inherent to the algebra of propositional logic. In short, reasoning is nothing more than the propositional calculus itself.**
>
> *Barbel Inhelder and Jean Piaget (1958), The growth of logical thinking from childhood to adolescence, New York: Basic Books.*

Propositional logic serves to help explain what logic and inference are but it is too inexpressive for most purposes. It deals only with unstructured propositions whereas we will need to represent relations between objects and various other extensions. So the search for the "true logic" continued:

> **As I awoke in the morning, the sun was shining brightly into my room. There was a consciousness on my mind that I was the discoverer of the true logic of the future. For a few minutes I felt such a delight such as one can seldom hope to feel.**
>
> *William Stanley Jevons (1866), quoted in W. Mays and D.P. Henry (1953), Jevons and logic, Mind, 62, 484.*

However, Jevons's logic proved another false dawn for it was only a variation on Boole's theme in which he used the 'inclusive or', which we defined above as standard for propositional logic today, instead of Boole's 'exclusive or', where `(p or q)` would be true only if `p` or `q` is true but not both. Still, Jevons did indicate the potential of propositional logic by constructing a 'logical piano' that was the first useful mechanical device for solving logical problems.

23. Predicate logic: "only one language suitable"

The most important step towards a useful symbolic logic was taken in 1879 by the German philosopher, Gottlöb Frege, who was again fully aware of his ambitious aim:

> **... a universal language in which every possible form of rational thought that could enter into a piece of deductive reasoning could be represented in a systematic and mathematically precise way.**
>
> *Gottlöb Frege (1879), Begriffschrift, in Peter Geach and Max Black (eds.), Translations from the Philosophical Writings of Gottlöb Frege, Oxford: Blackwell.*

He defined a logic, called predicate logic, that came to be the basis for almost all of the formal work in AI (and much else besides, of course). In predicate logic, the notation `P(a, b)` is used to express the fact that `a` is related to `b` by the predicate `P`. For example,

```
Killed(Brutus, Caesar)
```

might denote the meaning of the sentence "Brutus killed Caesar". Of course, as with propositional logic, any interpretation that we may put on an expression in predicate logic is up to us and is not part of the formal system. The logic itself is only concerned with the form of expressions and the definition of rules of inference.

Predicate logic uses connectives as in propositional logic, for example, in an expression such as

```
Roman(Caesar) and Married(Caesar, Clapurnia).
```

Predicate logic extends propositional logic in including variables, quantifiers and functions. A variable may be used in an expression such as

```
White(x)
```

interpreted perhaps to mean "All `x` are white" or "Everything is white". The `x` here is said to be universally quantified. (Variables may also be existentially quantified, to mean "Some `x` are white" but since this is equivalent to "Not all `x` are not white" this is not really necessary.) Used with connectives, variables enable the expression of complex general statements, for example

```
not Swan(x) or White(x)
```

might mean "For all `x`, either `x` is not a swan or `x` is white" or, equivalently, "All swans are white".

Functions are used to enable statements to be made about unnamed individuals or objects. For example,

```
Killed(husband(Lady Macbeth), Duncan),
```

might mean "The husband of Lady Macbeth killed Duncan". Indeed, much of the power of predicate logic derives not so much from what it enables us to say but from what it lets us leave unsaid. For example, we could write

```
not(Ring(Uranus, x) → Circular(x))
```

to be interpreted as "It is not the case that if `x` is a ring of Uranus then `x` is circular" or "Not all Uranus's rings are circular", without having to name the rings or say how many there are.

To permit conclusions to be drawn in predicate logic, there is a generalised version of the rule of resolution that we introduced above for propositional logic. An example of the use of resolution in predicate logic (with possible interpretations of the expressions indicated) is:

Premise 1: `M(x) → G(x)`
that is, in conjunctive normal form:
`not M(x) or G(x)` "All Martians are green"
Premise 2: `G(y) → E(y)`
that is, in conjunctive normal form:
`not G(y) or E(y)` "Everything green is edible"
Premise 3: `M(Marty)` "Marty is a Martian"
Inference (applying resolution to premise 1 and premise 3):
`G(Marty)` "Marty is green"
Inference (applying resolution to premise 1 and premise 2):
`not M(z) or E(z)` "All Martians are edible"

Whereas the propositional version of resolution looks for a `p` and a `not p` which can be 'cancelled out', in predicate logic we need two terms, such as `M(Marty)` and `not M(x)`, that can be made the negation of one another with a suitable substitution for variables.

There has been a vast amount of theoretical and practical work on the development of efficient computational techniques for proving theorems using resolution and on the development of extended notations and interpretations of predicate logic expressions, to the extent that predicate logic is now the standard form of representation of information within AI programs. For its enthusiasts, it is the *only* form of representation:

> **There is only one language suitable for representing information – whether declarative or procedural – and that is first-order predicate logic. There is only one intelligent way to process information – and that is by applying deductive inference methods.**
>
> *Robert Kowalski (1980), position paper in Special Issue on Knowledge Representation, SIGART Newsletter.*

In a 'first-order' logic, by the way, variables refer only to individuals or objects. They are not allowed to represent predicates or functions, so that sentences such as "All binary predicates are transitive" cannot be naturally expressed. Higher order logics are possible, but usable proof procedures do not exist for them.

The argument, then, is that predicate logic theorem proving may be used not just narrowly to prove mathematical theorems but also to form the core process of many or all AI problem solving activities. For example, we may apply theorem proving to robot planning problems. Premises would describe

the robot's world, that is, the location of objects, the dimensions of rooms, and so on. The robot's actions would be expressed by axioms defining how the world changes as a result of such actions. A problem such as developing a plan to, say, move an object from one place to another would then be transformed into one of proving that the desired state of the world can be achieved by means of a certain sequence of actions, to be discovered by the proof. As it stands, the theorem-proving process only says whether or not such a plan exists: a fairly simple extension (basically, to track the substitutions made to variables during the proof) enables the plan itself to be determined.

Another example is that of diagnosis, that is, the problem of obtaining an explanation for some observed and generally undesired or unexpected data. Say we are diagnosing some faulty equipment. We could write premises to define how all the components are supposed to work and how the components are put together (that is, how the outputs from some components form the inputs of others). Given the input to the equipment we could then derive, by theorem proving, the correct, expected output. However, the observed output is different, since the equipment is faulty. To explain this, we could systematically withdraw or change the premises describing the function and structure of components. If this edited description enables us to derive the observed data we have a potential explanation of it.

In case robot planning and fault diagnosis seem minor application areas, the sweeping nature of what is being proposed should be emphasised. Automatically deriving a plan is an instance of generating a program to achieve some goal. So it is being suggested that theorem proving may enable programs to generate, on-the-fly, other programs to achieve goals as they arise, the initial program therefore not being intrinsically limited in its capabilities. Automatic diagnosis is an instance of the general task of finding an explanation for some observed data, which is the core of the scientific discovery process. Theorem proving might, therefore, enable programs to discover theories for interesting observations.

Obviously, such applications of theorem proving need to be more precisely defined and their computational utility explored but it seems that we may have a technique of broad relevance to AI. However, the logicist's position – that we should express all our knowledge in a formal notation like predicate logic, to be processed by defined rules of inference – is subject to a number of criticisms. Here is a baker's dozen of them.

24. Reasoning in practice: "you cannot come to any conclusion"

The criticisms of automated reasoning can be loosely grouped into three classes: those of practice (that is, concerned with the difficulty of reasoning effectively); those of theory (that is, concerned with the theoretical limitations of the predicate logic system); and those of principle (that is, concerned with doubts about the relevance of the process). We will list these criticisms as dilemmas (for they are twice as interesting as the lemmas of serious mathematical texts).

Dilemma 1: Automated reasoning is too laborious and lacks the precision that seems to characterise human thought:

> **A calculus is a diversion one cannot afford; it is a combine harvester when we need a carving knife.**
>
> *Michael Scriven (1976), Reasoning, New York: McGraw Hill.*
>
> **A real proof is not checkable by a machine, or even by any mathematician not privy to the gestalt, the mode of thought of the particular field of mathematics in which the proof is located. Even to the "qualified reader", there are normally differences of opinion as to whether a real proof (i.e. one that is actually spoken or written down) is complete or correct. These doubts are resolved by communication and explanation, never by transcribing the proof into first-order predicate calculus.**
>
> *Philip J. Davis and Reuben Hersh (1981), The Mathematical Experience, Boston: Birkhäuser.*

However, experience with computer programming should have thoroughly disabused us of the notion that humans are reliably precise in their reasoning. Our ability to focus our reasoning in order to arrive at conclusions within a useful time is often at the cost of reliability. In most situations, both humans and computers need to 'leap to conclusions', even if this is at the cost of soundness. It is likely therefore that means will be needed to provide computers with the insight, for want of a word, to avoid a relentless grind through the premises.

Dilemma 2: Theorem-proving methods are computationally inefficient, with the result that even the most straightforward of conclusions may take an excessive time to prove:

> **Trust the man who hesitates in his speech and is quick and steady in action, but beware of long arguments and long beards.**
>
> *George Santayana (1922), Soliloquies in England, The British Character.*

Imagine any kind of realistic problem, with possibly thousands of potentially useful premises. Any two of these may be used with the rule of resolution to

derive a further expression, which can be used in further uses of resolution. How can any general-purpose scheme, like using the rule of resolution, select the best premises to derive the conclusion desired?

It is difficult to appreciate the distinction between computations that are possible 'in principle' and those possible 'in practice'. Say it takes 1 millisecond to derive a conclusion from 10 premises (which sounds quite acceptable) and that every extra premise doubles the time (which is quite possible). Then if we had 100 premises (which isn't that many) it would take over 10,000,000,000,000,000 years, which is about a million times longer than the universe has existed. Unfortunately, theorem proving and many other AI problems have just this property: the time goes up exponentially with the size of the problem. More efficient hardware cannot overcome this fact of life – to all intents and purposes, large problems of exponential complexity are not practically solvable.

Dilemma 3: First-order predicate logic is undecidable, which means that there will always be theorems which should be derivable from the premises but which cannot be proved in a finite number of steps with a systematic procedure like using the rule of resolution:

> **I have come to the conclusion, after many years of sometimes sad experience, that you cannot come to any conclusion at all.**
>
> *Vita Sackville-West (1953), In Your Garden Again, London: M. Joseph.*

So not only does predicate logic theorem proving take a long time but in some cases it would take an infinity!

The incompleteness theorem of Kurt Gödel, which shows that in any logical system worthy of the name statements can be formulated which can be neither proved nor disproved within the system, was proved in 1931 and had profound implications for the goal of mathematics to be able to prove anything provable – it showed the goal to be unattainable, and you can't get much profounder than that. The implications of Gödel's theorem for and against attempts to automate reasoning have continued to be the subject of much debate but its intrinsic irrelevance was summarised some time ago:

> **Gödel's conclusions bear on the question whether a calculating machine can be constructed that would match the human brain in mathematical intelligence. Today's calculating machines have a fixed set of directives built into them: these directives correspond to the fixed rules of inference of formalized axiomatic procedure. [But] the resources of the human intellect have not been, and cannot be, fully formalized ... mathematical propositions which cannot be established by formal deduction from a given set of axioms may, nevertheless, be established by "informal" meta-mathematical reasoning. It would be irresponsible to claim that these**

> **formally indemonstrable truths established by meta-mathematical arguments are based on nothing better than bare appeals to intuition. Nor do the inherent limitations of calculating machines imply that we cannot hope to explain living matter and human reason in physical and chemical terms. The possibility of such explanations is neither precluded nor affirmed by Gödel's incompleteness theorem.**
>
> *Ernest Nagel and James Newman (1958), Gödel's Proof, London: Routledge and Kegan Paul.*

Indeed, Alan Turing had in his 1950 paper that introduced the idea of the Turing test already tried to discount the Gödel theorem objection to the possibility of AI.

Nonetheless, it continues to be invoked. For example, Roger Penrose, who was awarded the 1988 Wolf Prize for physics for his work with Stephen Hawking on black holes, wrote *The Emperor's New Mind* (which AI researchers, apparently mistakenly, took to be a suggestion that their new theories were mindless) in which he used Gödel's theorem to argue that, since mathematicians can 'see' a theorem to be true even if it cannot be proved, mathematical argument needs 'consciousness' (a notion to be returned to later):

> **It seems to me that it is a clear consequence of the Gödel argument that the concept of mathematical truth cannot be encapsulated in any formalistic scheme … Mathematical truth is *not* something that we ascertain merely by use of an algorithm. I believe, also, that our *consciousness* is a crucial ingredient in our comprehension of mathematical truth … When one 'sees' a mathematical truth, one's consciousness breaks through into this world of ideas, and makes direct contact with it. When mathematicians communicate, this is made possible by each one having a *direct route to truth*.**
>
> *Roger Penrose (1989), The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics, Oxford: Oxford University Press.*

Penrose himself, being an eminent mathematician, clearly has this hotline to truth, so one can hardly argue with that. However, even if the argument is sound, it seems to apply only to mathematicians. It does not prove that the thinking of the rest of us "cannot be encapsulated in any formalistic scheme". Maybe it is assumed that our thinking is messier than that of mathematicians and is therefore even less likely to be formalisable.

However, the philosopher Hilary Putnam dismisses Penrose's later book *Shadows of the Mind*, which repeated the argument of *The Emperor's New Mind*, as:

... a sad episode in our current intellectual life … [Penrose] is convinced by an argument that all experts in mathematical logic have long rejected as fallacious … He mistakenly believes that he has a philosophical disagreement with the logical community, when in fact it is a straightforward case of a mathematical fallacy.

Hilary Putnam (1994), New York Times review of Roger Penrose (1994), Shadows of the Mind: A Search for the Missing Science of Consciousness, Oxford: Oxford University Press.

The argument that Putnam considers Penrose to rely upon was made by John Lucas, an Oxford philosopher, in 1961. However, the set of 'all experts' that Putnam says have long rejected the argument does not include Lucas himself, who continues to defend his position. These contradictory views of experts might lead us to conclude that it would be unwise to dismiss the possibility of AI on the basis of Gödel's theorem alone.

Dilemma 4: The method assumes a complete set of necessary premises and these are rarely available:

Life is the art of drawing sufficient conclusions from insufficient premises.

Samuel Butler (1912), Note-Books, "Lord, what is man?"

Samuel Butler (1835-1902), the *Note-Books* of whom were published posthumously, is best known today for his satire *Erewhon* (1872), in which, amongst other things, the development of machinery is banned after it threatened to usurp human supremacy and led to civil war. His aperçu above is particularly apt for theorem proving using resolution because, obviously, the method depends upon all the required premises being available because, as it stands, there is no way to detect and fill apparent gaps in what is needed to complete a proof.

Dilemma 5: The method assumes that the set of premises is consistent, which is difficult to ensure:

"Logical" reasoning is not flexible enough to serve as a basis for thinking. I prefer to think of it as a collection of heuristic methods, effective only when applied to starkly simplified schematic plans. The consistency that logic absolutely demands is not otherwise usually available – and probably not even desirable! – because consistent systems are likely to be too "weak".

Marvin Minsky (1975), A framework for representing knowledge, in Patrick Winston (ed.), The Psychology of Computer Vision, New York: McGraw-Hill.

Consistency is contrary to nature, contrary to life. The only completely consistent people are the dead.

Aldous Huxley (1929), Do What You Will, London: Chatto & Windus.

Like Samuel Butler, Aldous Huxley is best known today for a satirical novel that can also be seen as a protest against increased mechanisation. His *Brave New World* (1932) explored the incompatibility of individual freedom and a scientifically controlled society.

A set of premises is inconsistent if it is possible to derive both `q` and `not q` from them. Using resolution with inconsistent premises we would be able to derive a contradiction regardless of the particular conclusion we were trying to derive and so apparently succeed in proving any expression whatsoever to be a theorem:

One falsehood spoils a thousand truths.

Proverb of the Ashanti of Ghana.

If the set of premises is inconsistent then, in the simplest case, it would include a proposition `p` and its negation, `not p`, and clearly one of them should be deleted from the set, although we would not necessarily know which one. In general, though, the contradiction would be more indirect and it would be difficult to pinpoint the 'cause' of the internal inconsistency – but at least we would know there is one.

Putting a positive gloss on the need for consistent premises, if by applying the rule of resolution repeatedly to a set of premises, which are assumed to be true, we find that the set of premises is in fact inconsistent then at least we have some basis for revising our knowledge, as expressed in the premises:

In formal logic, a contradiction is the signal of defeat: but in the evolution of real knowledge it marks the first step in progress towards a victory.

Alfred North Whitehead (1925), Science and the Modern World, New York: Simon and Schuster.

In other words, we might have the basis for a method for a computer to improve its knowledge, that is, to learn, as we will consider later.

25. Reasoning in theory: "a deep epistemological problem"

The second class of criticism is concerned with the theoretical difficulties of predicate logic.

Dilemma 6: Predicate logic, being a timeless notation, cannot easily handle notions of time and related concepts, such as causality. For example, a logical expression such as

```
Fell-down(Jack) and Broke-his-crown(Jack)
```

could equally well mean "Jack fell down and (then) broke his crown" or "Jack broke his crown and (then) fell down". The usual attempt to overcome such problems is to time-stamp the terms, as in

```
Fell-down(Jack, t1) and Broke-his-crown(Jack, t2)
```

where `t1` is before or after `t2`, as appropriate. Actually, the time-stamps are generally considered to denote situations (`s1`, `s2`, and so on) of the world. To say that there is some causal relation between the two situations, we might attempt something like

```
Ok(Jack, s1) and Falls-down(Jack, s1)
   → Broken-crown(Jack, fallen-down(Jack, s1))
```

that is, "If Jack is OK in situation `s1` and falls down, then in the situation of the world reached by his having fallen down he has a broken crown".

This may seem to be getting complicated enough but it is not a fraction of what we need. What about his companion Jill? What can we say about her, in the new state of the world reached by Jack having fallen down? The answer is nothing at all unless we write some premises to say, perhaps, that whatever was true about Jill in the initial situation is still true in the new situation. But, of course, this may well not be the case – she may now be distraught or delighted at Jack's injury, or she may have fallen down herself. The same goes for every other individual in the universe! We seem to need a huge number of premises to be able to reason about what changes, and what does not change, in the world.

In short, logic does not easily capture the inertia of the real world, that is, the fact that, as the world changes, most things remain unchanged:

How awful! Do you still have an artificial leg?

Simon Fanshawe, in a radio interview after the interviewee had said that her most embarrassing moment was when her artificial leg fell off at the altar on her wedding day.

The problem of reasoning about what changes and what does not change has proved sufficiently taxing to earn its own name, the frame problem. The most elegant answer appears to lie in defining, for each predicate that may change its value over time, precisely how it may become true or untrue – but it remains hardly a trivial matter.

Opinions on the significance of the frame problem vary widely. On the philosophical side, some regard it as just a re-working of issues that philosophers have mulled over before, while others think:

… on the contrary, that it is a new, deep epistemological problem – accessible in principle but unnoticed by generations of philosophers – brought to light by the novel methods of AI, and still far from being solved.

Daniel Dennett (1984), Cognitive wheels: the frame problem of AI, in Christopher Hookway (ed.), Minds, Machines, and Evolution, Cambridge, Mass.: Cambridge University Press.

Computationally, some think that it was a minor technical difficulty now essentially solved for all practical purposes, while others consider it to be a fundamental impasse of such magnitude that it means that AI is impossible.

Dilemma 7: Predicate logic is a two-value (true-false) system, making it difficult to express the vagueness of ordinary facts and to draw conclusions that are plausible but perhaps not certain:

Indeed, you can build a machine to draw demonstrative conclusions for you, but I think you can never build a machine that will draw plausible inferences.

George Polya (1965), Mathematical Discovery, New York: John Wiley & Sons.

I have known uncertainty: a state unknown to the Greeks.

Jorge Luis Borges (1945), Ficciones, The Babylonian Lottery, New York: Grove Press.

There is a sense of uncertainty implicit in the disjunction `p v q`, indicating that `p` or `q` (or both) is true but we are not certain which. However, we cannot be precise about this uncertainty. We cannot say, for example, that one proposition is twice as likely to be true than the other or reason directly with these probabilities. An unembellished predicate logic does not deal in degrees of certainty.

An analysis of certainty is something that we might reasonably have expected from logic. If we are to go to all the trouble of expressing our knowledge in some logical notation then the least we might ask for in return is some indication of the reliance that we may place on the conclusions derived. Indeed, we might believe that it is our aim to acquire reliable or certain knowledge, and that a rigorously defined procedure for determining it will help, and that if we persevere we are surely bound to attain certain knowledge:

In time you will know all with certainty.

Sophocles (427 B.C.), Oedipus the King.

However, as we have seen, logic cannot deliver certain knowledge. Much as we may be in awe of logic, it is concerned only with the validity of the process of obtaining a conclusion from a set of premises. The conclusion cannot be considered 'certain knowledge' and, in any case, certainty may be an unwise objective:

We can be absolutely certain only about things we do not understand.

Eric Hoffer (1951), The True Believer: Thoughts on the Nature of Mass Movements, San Francisco, Calif.: HarperCollins.

The one unchangeable certainty is that nothing is certain or unchangeable.

John F. Kennedy (1962), State of the Union Address.

Like Kennedy's comment, that of Hoffer was made in a political context. His book, which contended that 'the true believer' comes from the ranks of those who need a cause to substitute for their lost faith in themselves, is being reassessed as foretelling recent world events. It also made the point, which relates to the recent trend towards collaborative computing, that thinkers rarely work well together, whereas people of action readily build camaraderie.

Dilemma 8: Predicate logic has some formal properties that may make it unsuitable for expressing what we would like to express. For example, first-order predicate logic is what is called referentially transparent, which means that the replacement of any term by an equivalent term is allowed. With the obvious interpretation of symbols, we would consider the two terms

```
Beethoven
composer(Fidelio)
```

to be equivalent. So we could re-write an expression such as

```
German(Beethoven)
```

as

```
German(composer(Fidelio)).
```

However, it is possible for someone to believe that Beethoven was German without necessarily believing that the composer of Fidelio was German (Mary might believe Mozart composed Fidelio) and therefore we should not be allowed to re-write

```
Believes(Mary, German(Beethoven))
```

in the form

```
Believes(Mary, German(composer(Fidelio)).
```

And indeed we are not allowed to re-write it – because we are not allowed to write it: in first-order predicate logic we cannot write propositions within propositions, as we have just tried to do. That still leaves us, of course, with the problem: just how do we express the meaning of a sentence such as "Mary believes Beethoven was German" in logic? The

```
Believes(Mary, German(Beethoven))
```

expression is actually one within a 'modal logic', for which different proof procedures need to be devised.

A vast variety of modal logics have been defined to try to capture the nature of belief and knowledge, as well as other concepts, such as time and necessity. A modal logic augments predicate logic with modal operators that apply to logical sentences. For example, a modal operator might denote that "It is believed that …", "It will always be true in the future that", or "It is necessarily the case that …".

We also need a set of inference rules to deal with the modal operators. For example, in a modal logic of knowledge, we might try

$$\frac{\texttt{Knows(a, p)}}{\texttt{Knows(a, Knows(a, p))}}$$

that is, if somebody knows something then they know that they know it:

> **A person who knows anything, by that very fact knows that he knows and knows that he knows that he knows, and so on ad infinitum.**
>
> *Benedict de Spinoza (1677), Ethics, II, Prop. 21.*

Spinoza's *Ethics* was written in 1665 but withdrawn because his comments on the scriptures were controversial and only published after his death. It included a recommendation for systematic reasoning, in which true ideas were to be expressed in definitions, from which further true propositions may be deductively derived. His contention that if we know something then we necessarily have complete reflective self-knowledge of that fact will be happily debated by philosophers ad infinitum, and quite reasonably so because doubts can be held about this and every other rule of inference that has been suggested for modal logics. Particularly controversial is a rule of inference that says that we know everything implied by what we know (a property called 'logical omniscience'):

$$\frac{\texttt{Knows(a, p) and Knows(a, p} \rightarrow \texttt{q)}}{\texttt{Knows(a, q)}}$$

26. Reasoning in principle: "I know it by my heart"

The third class of criticism is concerned with objections to the principle of automated reasoning itself, considering it to be somehow misguided or irrelevant.

Dilemma 9: The underlying assumption that intelligence derives from the conscious or unconscious processing of propositions is unsound:

> **The general assertion that all intelligent performance requires to be prefaced by the consideration of appropriate propositions rings unplausibly, even when it is apologetically conceded that the required consideration is often very swift and may go quite unremarked by the agent ... the intellectualist legend is false and ... when we describe a performance as intelligent, this does not entail the double operation of considering and executing … It is therefore possible for people intelligently to perform some sorts of operations when they are not yet able to consider any propositions enjoining how they should be performed.**
>
> *Gilbert Ryle (1949), The Concept of Mind, New York: Barnes and Noble.*

The deliberative-reactive debate that we discussed in the context of planning echoes Ryle's distinction between "considering and executing", which he made as part of a general attack on the traditional metaphysical dualism of mind and body. He argued that the dogma of what he called "the ghost in the machine", that is, the mind in the body, was mistaken. He considered that mental concepts (such as the taste of a pineapple or the idea of an isosceles triangle) referred not to ghostly acts on internalised entities but to dispositions to behave in certain ways in certain situations. As such, his ideas are precursors to those of the situated cognition movement.

Dilemma 10: As a psychological model of human reasoning the method is unrealistic:

> **A mind all logic is like a knife all blade. It makes the hand bleed that uses it.**
>
> *Rabindranath Tagore (1916), Stray Birds, 193, New York: MacMillan.*

(This profound comment reminds me of the warning given with a knife sold by Olfa Corporation: "Caution: blade is extremely sharp. Keep out of children.").

It is rather odd to criticise formal logic for being unlike human reasoning when the former has presumably been developed precisely to help overcome the limitations of the latter. In any case, it is not necessary that an artificial intelligence reasons in the same way as a human intelligence if the aim is only to achieve some level of reasoning performance. It is natural to look towards the only existing reasoner for inspiration, just as early car designers based their signalling devices on the only existing signaller, by adding two little arms which flapped out of the sides of cars: it was several decades before winkers at each corner were accepted as better. However, for many researchers the main purpose of trying to build artificial reasoners is to gain insights into the way the only other known reasoner, the human reasoner, reasons. While some may be unable to conceive that an artificial reasoner could possibly not be modelled on a human reasoner, maybe it is better to view artificial reasoning as complimentary to, and not a simulation of, human reasoning:

> **The power of logic and mathematics to surprise us depends, like their usefulness, on the limitations of our reason.**
>
> *Alfred J. Ayer (1936), Language, Truth and Logic, New York: Dover Publications.*

Clearly, artificial reasoners are not subject to the same difficulties, such as memory limitations, as humans.

Dilemma 11: The precise symbolisation demanded by computational logic renders it inapplicable to the problems of real life. It is just much too

difficult to express anything worth the effort in a notation such as predicate logic:

All traditional logic habitually assumes that precise symbols are being employed. It is therefore not applicable to this terrestrial life but only to an imagined celestial existence ... logic takes us nearer to heaven than other studies.

Bertrand Russell (1923), Vagueness, Australian Journal of Philosophy, 1, 84-92.

As Russell implies, by taking us to heaven, logic is of no earthly use.

Dilemma 12: Reasoning is a subtle, mystical, intuitive process that cannot be satisfactorily reduced to elementary rationalism:

Intuitive conviction surpasses logic as the brilliance of the sun surpasses the pale light of the moon.

Morris Kline, ed. (1968), Mathematics in the Modern World: Readings from Scientific American, San Francisco: W.H. Freeman.

To know, to get into the truth of anything, is ever a mystic net, of which the best logics can but babble on the surface.

Thomas Carlyle (1841), On Heroes, Hero-Worship and the Heroic in History.

Whatever logics Thomas Carlyle was referring to in 1841, modern logics such as predicate logic with rules of inference such as resolution are really quite simple formalisms, as mathematical formalisms go, and it may seem implausible that they are capable of capturing the essence of the supposedly unique human ability to reason. Intuition, insight, mysticism – yes, we might well suspect that they surpass logical reasoning but what exactly are they, that computers may (or may not) be endowed with them? Actually, researchers in the area concede that there may well be more to reasoning than is captured in current automated processes:

Reasoning is an art and not a science …

Larry Wos, Ross Overbeek, Ewing Lusk and Jim Boyle (1984), Automated Reasoning: Introduction and Applications, Englewood Cliffs, N.J.: Prentice-Hall.

However, the acceptance that automatic methods are limited does not mean that they have no use at all.

Dilemma 13: Even if all the above objections could be overcome, nonetheless the whole exercise somehow misses the point:

We can prove whatever we want to, and the real difficulty is to know what we want to prove.

Alain, the nom de plume of Émile-Auguste Chartier (1920), Le Système des Beaux Arts.

"I and all people have only one firm, unquestionable and clear knowledge, and this knowledge cannot be explained by reason – it is outside it, and has

no causes, and can have no consequences … what I know, I do not know by reason, it is given to me, it is revealed to me, and I know it by my heart."

Leo Tolstoy (1877), Levin, in Anna Karenina.

That which needs to be proved cannot be worth much.

Friedrich Nietzsche (1889), Götzendammerung, oder: Wie man mit dem Hammer philosophirt (Twilight of the Idols, or How One Philosophizes with a Hammer).

Amongst the famously abstruse diatribes of Nietzsche on democracy, religion, morality, altruism and much else besides there seems to be an argument that rationalism, as pursued in western philosophy since Socrates, is decadent because it places reason and instinct in opposition. Without affirming the primacy of the latter, he wanted some integration of reason and instinct, along with passion and the will, but unfortunately and inevitably he was unable to specify in any detail how this might be achieved.

We began the discussion on reasoning in the hope that it would provide the key to intelligence. If we could only automate the processes of reasoning we would have the basis for intelligent computers. Now, it seems, as soon as we begin to feel that automated reasoning may be within our grasp, we are reminded that it had already been said that it would be worthless, irrelevant and impossible.

27. Common sense: "a wild thing"

The weight of criticism seems overwhelming, just when we seemed to be making progress in fulfilling the dreams of Leibniz and others:

I was upset. I had always believed logic was a universal weapon, and now I realized how its validity depended on the way it was employed.

Umberto Eco (1983), The Name of the Rose, London: Martin Secker and Warburg.

It seems that I have spent my entire time trying to make life more rational and that it was all wasted effort.

Alfred J. Ayer (c1980).

The disillusionment of Ayer is particularly pointed, for his *Language, Truth and Logic* of 1936 had done the most to bring logical positivism to the English-speaking word from the Vienna Circle and he was coming to his conclusion that its avowal of the benefits of rationalism was mistaken at just the time that AIers were committing themselves to a programme to make it work.

The force of the criticism needs to be balanced against the fact that AI is continually engaged in the activity of redefining what a 'logic' is. There is now a large range of non-standard logics (temporal logics, modal logics,

deontic logics, non-monotonic logics, default logics, and so on) and mathematical books on 'commonsense reasoning', which to a layperson might seem a self-contradiction.

John McCarthy, one of the initiators of the 1956 Dartmouth conference, has pursued a lifelong quest to formalise common sense so that computer programs might make use of it. His Advice Taker proposal of 1959 has motivated almost all of his subsequent research contributions. The Advice Taker system was intended to be able to take advice from a person in order to improve its performance, making any necessary commonsense inferences automatically:

> **This property is expected to have much in common with what makes us describe certain humans as having common sense. We shall therefore say that a program has common sense if it automatically deduces for itself a sufficiently wide class of immediate consequences of anything it is told and what it already knows.**
>
> *John McCarthy (1959), Programs with common sense, in the Proceedings of the Teddington Conference on the Mechanization of Thought Processes, 75-91.*

The key word here is "immediate". As we have seen, automatic theorem proving tends to be slow. We would like a computer to somehow make 'obvious' inferences instantaneously, without, of course, making too many inferences that are risky or are unlikely to be needed. The difficulty of achieving this is clear from considering any everyday example. For example, what does your common sense indicate would be the sufficient, immediate consequences of a news item such as "A severe tornado hit Atlanta yesterday"? (Well, I hope this isn't an everyday example.)

By definition, common sense is possessed by us all:

> **Common sense is the most widely distributed commodity in the world, for everyone thinks himself so well endowed with it that those who are hardest to please in any other respect generally have no desire to possess more of it than they have.**
>
> *René Descartes (1637), Le Discours de la Méthode.*

It may indeed seem to be something upon which we can put our trust, without needing formal logics and the like:

> **Do not be bullied out of your common sense by the specialist; two to one, he is a pedant.**
>
> *Oliver Wendell Holmes, (1891), Over the Tea-Cups.*

Unfortunately, what is common to the human species does not necessarily easily extend to computers – in fact, to the contrary, because common sense seems to come so naturally to us we easily overlook just how complex and

powerful it is. Indeed, because of the ubiquity of common sense, it is also jostling for a central position in the sphere of intelligence:

> **… most of what we know and most of the conscious thinking we do has its roots in common sense. Thus, a complete theory of common sense would contain the fundamental kernel of a complete theory of human knowledge and intelligence.**
>
> *Ernest Davis (1990), Representations of Commonsense Knowledge, San Mateo, Calif.: Morgan Kaufmann.*

But before considering attempts to develop such a theory, it should be emphasised that 'commonsense reasoning' is not to be regarded as a kind of inferior reasoning carried out by ordinary people without skills in 'expert reasoning'. Commonsense reasoning is usually taken to mean reasoning (by all, including specialists) that efficiently reaches useful conclusions that perhaps do not strictly follow from what is known. Ordinary logic assumes or aims for complete and certain knowledge whereas commonsense reasoning takes account of all sorts of assumptions and uncertainties in order to come to some effective conclusions within a reasonable time.

28. Plausible reasoning: "controversial and provisional"

George Polya distinguished between demonstrative reasoning (the kind exemplified by resolution-based proving) and plausible reasoning:

> **We secure our knowledge by demonstrative reasoning, but we support our conjectures by plausible reasoning … Demonstrative reasoning is safe, beyond controversy, and final. Plausible reasoning is hazardous, controversial and provisional … Anything new that we learn about the world involves plausible reasoning, which is the only kind of reasoning for which we care in everyday affairs.**
>
> *George Polya (1954), Mathematics and Plausible Reasoning, Princeton: Princeton University Press.*

George Polya (1887-1985) pioneered the study of problem solving, with his 1945 book *How to Solve It* selling over a million copies.

The approaches in AI to plausible reasoning can be distinguished as logic-based or probability-based. The logic-based methods attempt first to tackle the problem that first-order predicate logic is 'monotonic', that is, if you can prove something from a set of premises then you can still prove it if you add further premises, whereas most commonsense reasoning is non-monotonic. For example, if you read the word "cuttlefish" and do not know what it is, you might reasonably conclude that it is a fish, that is, a cold-blooded vertebrate with gills and fins. If you are later informed that a

cuttlefish is a mollusc then (if you have an idea what a mollusc is) you will happily withdraw some of your earlier conclusions.

The approach of withdrawing conclusions on the basis of new information received forms the AI field of 'belief revision', which does not denote a religious conversion but is concerned with changing a knowledge base, by considering the support for the contents, in order to get rid of a detected contradiction. As regards plausible reasoning, however, it is better to think in terms of the extra information blocking deductions that were previously possible.

The various logics for plausible reasoning bask in luxurious nomenclature and forbidding theoretical undergrowth. However, since they are intended to capture some aspect of commonsense reasoning we should be able to express their basic ideas in commonsense language. The 'closed-world assumption' is that all the significant relations between entities are known, and therefore if something is not known or cannot be shown to be true then it must not be the case. For example, if we are told that America, France, Britain, India and Israel have the nuclear bomb, then we might deduce that Australia does not, the assumption being that it would have been mentioned if it had. The technique of 'circumscription' (which exists in multiple flavours: equality, formula, parallel, prioritized, pointwise, and so on) is a more general and powerful version of the closed-world assumption. The idea is that if some property typically holds then we minimise the set of exceptional individuals, that is, ones that lack this property. An 'autoepistemic logic' makes inferences along the lines of "if something were true then I would know it, and I don't, so it cannot be". For example, if someone were to claim to be your brother then you might deny it since if he were then you would have known about him (unless you are a character in a soap opera). These logics, then, provide a set of techniques based on the assumption that we are told everything that is relevant.

Another approach is based on making default assumptions. Consider, for example, the sentence "A person is innocent unless proved guilty". We could try to express this in ordinary logic as

```
not Proved-guilty(x) → Innocent(x).
```

This does not, however, capture the default assumption that the person is innocent:

In former days, everyone found the assumption of innocence so easy; today we find fatally easy the assumption of guilt.

Amanda Cross (2001), Poetic Justice, Fawcett Books: New York.

The predicate logic expression requires us to fail to prove him or her guilty, leaving the burden of proof with the defence, not the prosecution. We could add

```
not Innocent(x) → Proved-guilty(x)
```

but that would regard innocence and guilt equally. Default logics extend classical logic by adding inference rules of the form "if `p` is true and I cannot show `q` then infer `r`". For example, if we hear about a cuttlefish and have no evidence that it is not a fish then we might infer that it has fins. Of course, this is a form of stereotypical reasoning, which has its dangers as well as its benefits. It corresponds to inferences such as "Surgeons tend to be male; I have no evidence that Dr Smith is not male; therefore I will assume Dr Smith is male."

In addition to such general-purpose plausible reasoning schemes, there are many special-purpose notations intended to deal with common concepts (such as time and space), with specific subject areas (such as physics), and with individual and group conceptions. For example, to enable a computer to reason about time, we need to make some assumptions about the nature of time in order to define a temporal logic, for computers lack our natural aptitude for temporal reasoning:

"So, did you see which train crashed into which first?"

"No, they both ran into each other at the same time."

Extract from a radio interview.

Notwithstanding the apparently instantaneous nature of such an event, James Allen, taking his inspiration from elsewhere:

Vladimir: That passed the time.

Estragon: It would have passed in any case.

Vladimir: Yes, but not so rapidly.

Samuel Beckett (1955), Waiting for Godot.

argued that the time interval, rather than a point of time, was the best primitive in terms of which to define such a logic. An event is considered to have a beginning and (later) an ending. If we consider two events (with beginnings and endings of `b1`, `e1` and `b2`, `e2`, respectively) then each pair `[b1, b2]`, `[b1, e2]`, `[e1, b2]` and `[e1, e2]` can be related in one of three ways – the first may happen before, at the same time as, or after the second. So, in all, there are 3^4 (= 81) possibilities. It turns out that only 13 of these are possible in the real world. Maybe that reveals that the universe is inherently unlucky. Anyway, it is possible to use these 13 relationships to develop first-order logic axioms to reason about time.

Others prefer to derive alternative theories from different ontological commitments:

Time is the longest distance between two places.

Tennessee Williams (1945), The Glass Menagerie, New York: Random House.

But not all these points of view have been axiomatised.

The logic-based approaches to plausible reasoning are precisely defined by logicians and provide endless enjoyment to those investigating their overlapping properties. We merely note that many of them seem to involve some higher-level or recursive aspect: they seem to involve reasoning with a knowledge base to show, for example, that something does not follow from it, and then making an inference on that basis. On the surface, this seems paradoxical, for the first step (showing that something does not follow) is itself a kind of inference, so why does that not recurse to a yet higher level? To be effective and efficient, this first step somehow has to be different and faster than normal inference, as we would expect since we are trying to capture the notion of 'jumping to conclusions', not bury it under intractable formalisations. It is easy to see that this raises fascinating theoretical and practical questions, for those so inclined. At least, we may plausibly conclude that commonsense reasoning is not so trivial after all and its outcomes are not perhaps as reliable as we might wish:

Common sense ... has the very curious property of being more correct retrospectively than prospectively.

Russell Lincoln Ackoff (1968), in A. DeReuck, M. Goldsmith et al (eds.), Decision Making in National Science Policy, London: Ciba Foundation.

Common sense is a wild thing, savage, and beyond rules.

G.K. Chesterton (1903), Charles Dickens: A Critical Study, London: Hodder and Stoughton.

G.K. Chesterton is yet another writer to have written a satirical novel, *The Napoleon of Notting Hill* (1904), attacking the spread of technology. He is better known today for his *Father Brown* stories, about a priest who detected crime by intuitive methods rather than logical reasoning.

Another approach to overcoming the practical problems of resolution-based theorem proving is based on the realisation that some of the difficulties are due to the sheer number of premises, giving rise to all sorts of intermediate conclusions that are irrelevant to the theorem under proof. It would clearly be more efficient to limit reasoning to only those premises that are relevant, if only the system could tell what they are. A significant research effort is underway to determine means of establishing a context for 'local reasoning'. (The importance of context is indicated by the number of politicians and authors who complain that they have been quoted 'out of context', as if it could be otherwise. It is impossible to duplicate the original environment. Quotations are used to support the narrative they're in, not therein.) For

example, we could reason about John F. Kennedy as an American patriot, or as a philanderer, or as a Roman Catholic, and so on. It is entirely possible that conclusions reached within one context may contradict those reached in another, but as long as the system is restricted to reasoning within only one context at a time it will not be aware of such contradictions.

Or, to put it another way, the system may have a set of premises that, considered in its entirety, is inconsistent but this causes no problems as long as the system restricts its reasoning to a consistent subset. The modes of reasoning within different contexts may differ: they may be based on conventional resolution or they may use any of the plausible reasoning schemes mentioned above.

It is through the notion of context that logicians try to get a formal grip on the idea of situatedness. Instead of having expressions such as

```
At(table6, place234)
```

to mean that table6 is at place234, situated robots might use expressions such as

```
At(table6, 'in front of me').
```

Such expressions are said to be 'indexical', meaning that they depend on the robot's situation. Logicians and linguists have invested considerable effort in trying to deal with indexicals, which are widespread in natural language, for example, in sentences such as "I will complete this paragraph tonight", where the interpretation of "I", "this paragraph" and "tonight" depends on who uttered the sentence and when it was uttered. For the sake of completeness, we might consider "will complete" to be indexical too, as I may be much more thorough than you.

Related research focuses on 'relevance reasoning', in which the system reasons about which parts of its knowledge are relevant to a particular query. As commented above, this is a form of meta-reasoning, as it involves reasoning *about* the knowledge, rather than *with* it. There are also links with the various forms of non-monotonic reasoning mentioned above, as might be expected as they are all trying to capture the notion that there are typical cases, which form the basis for default reasoning, and non-typical cases, which may or may not be relevant.

Another way to try to provide practical logic-based reasoning is to relinquish the completeness seen as a virtue of resolution methods. After all, we do not make all possible inferences when answering queries – we make simple, superficial inferences and hope that they are adequate. One formalisation, introduced by Hector Levesque, involves distinguishing between explicit knowledge (that which a system knows directly or can infer immediately, say, by one inference step) and implicit knowledge (that which,

given sufficient time and motivation, a system could uncover by more persistent inference procedures). Such methods are still sound, of course, but no longer complete.

Yet another technique, knowledge compilation, is based on the fact that many of the limitations of automatic theorem proving mentioned above assume that we have arbitrarily complex logical expressions whereas in practice they may all be of a simpler form. Knowledge compilation tries to convert all the expressions into so-called Horn sentences, which are of the form

```
(p1 and p2 … and pn → q)
```

With Horn sentences inference is much faster than it can be with predicate logic expressions of unrestricted form. As much as possible of the knowledge base – which is often most of it, as Horn sentences are a natural form for many premises – is converted into Horn sentences before any inferences are made.

In summary, then, there is no shortage of ideas and effort to overcome the limitations of logic-based reasoning, thereby retaining the benefits of using a standardized, powerful notation. AI journals and conferences continue to publish more papers on the topic than on any other.

29. Probabilistic reasoning: "a revolutionary impact"

The logic-based approaches to plausible reasoning have a black and white view of propositions: they are considered to be true or false, but their relative strength is implicit in the way that they are variably corrigible. In other words, some propositions are much harder to block or discard than others but there is no explicit indication of the strength or probability of a proposition:

> **Belief is an all-or-nothing affair ... it is too complicated for mere finite beings to make extensive use of probabilities.**
>
> *Gilbert Harman (1983), Change in View: Principles of Reasoning, Cambridge, Mass.: MIT Press.*

However, it is possible to define 'multi-valued logics' in which propositions may take one of more than two values (say, true, false, and unknown). In the extreme, we may allow an infinity of values (say, any real number between 0 and 1, this number serving as a kind of probability that the proposition is true).

In fuzzy logic, such a number is used to indicate the degree to which the proposition is true rather than its probability, it being argued that many properties, for example, baldness and intelligence, are more-or-less true, rather than definitely true or false:

Fuzzy Wuzzy was a bear; Fuzzy Wuzzy had no hair
Fuzzy Wuzzy wasn't fuzzy, was he?

anon, in Louis Untermeyer, ed. (1959), Golden Treasury of Poetry, New York: Golden Press.

So, for example, an expression such as

```
bald(Fred, 0.9)
```

might indicate than, on a scale of baldness, Fred is very bald but not completely so (which is not the same thing as saying that the probability of Fred being bald is 0.9, as the expression would in a probabilistic logic). Fuzzy logic is rather sniffily disregarded by most AIers although it has more adherents in oriental countries, where perhaps the culture is less fixated on a philosophical tradition extolling the true/false distinction. The topic of vagueness, which is a challenge to classical logics that assume propositions to be true or false, remains hotly debated by philosophers of logic and language.

In probability-based approaches, the strength of a proposition is indicated explicitly by associating a number in the range 0 to 1 to indicate the likelihood that the proposition is true. Unlike fuzzy logic, it is assumed that the proposition is really true or false but we are uncertain which. So, instead of inferring, by default, that a cuttlefish has fins, we would indicate our expectation that this is the case by an expression such as

```
Fins(cuttlefish, 0.9).
```

On being told that a cuttlefish is not a fish, we might change this to, say,

```
Fins(cuttlefish, 0.05)
```

Reasoning in probabilistic logics is usually based upon a theorem developed in 1763 by Reverend Thomas Bayes. A friend of his fortunately recognised the significance of this theorem on looking through Bayes's papers after he died:

I now send you an essay which I have found among the papers of our deceased friend Mr Bayes, and which, in my opinion, has great merit.

Richard Price (1764), letter to the Royal Society of London.

The theorem says that, for example, the probability of a disease given a symptom, which is written P(disease|symptom), is related to the probability of the symptom given the disease and the probabilities of the disease and of the symptom as follows:

P(disease|symptom) = P(disease) x P(symptom|disease) / P(symptom)

A doctor can generally give reliable estimates for the probabilities on the right-hand side of this equation, for example, that the probability that a patient with a cardiac arrest has chest-pain is, say, 0.9, enabling what is usually required, the probability that a patient with chest-pain actually has a cardiac

arrest, to be determined. While Bayes' theorem (which tradition demands be spelled as Bayes' not Bayes's) may be mathematically sound, it does not entirely correspond to people's qualitative reasoning, which tends to rely on judgemental heuristics rather than statistical principles.

Nonetheless, in a formalism mainly due to Judea Pearl, Bayesian techniques have been developed to maintain the probabilities of networks of propositions. Imagine that we have a (very small!) Bayesian network representing that A causes B, A causes C, C causes D, and E causes D, or, if you prefer a specific example, angina causes breathlessness, angina causes chest-pain, chest-pain causes dizziness, and epilepsy causes dizziness. A doctor could attach probabilities to these causal links, that is, to the probability that if someone has angina they are breathless and, conversely, the probability that if someone is breathless they have angina (and also the probability that someone who does not have angina is breathless, and conversely), and so on for all the links. We can attach a priori probabilities to the nodes of the network, for example, the probability that someone about whom we have no information has angina.

Now, if we receive some information, say, that the patient has chest-pain, then we can re-calculate the probabilities for angina, breathlessness, dizziness, and so on, through the network. Intuitively, we would expect some of these probabilities to increase. Bayesian techniques give precise methods for rippling the effect of some evidence about one node through to all associated nodes. It is not necessary that the links between nodes represent causality, a philosophical hornet's nest: it just needs to be the case that evidence about one item enables the system to revise its probabilities about the associated item.

In practice, a Bayesian network may contain hundreds or thousands of nodes. If its establishment seems a daunting task then it should be borne in mind that much of the power of the techniques comes from the fact that the large majority of potential links are missing from the network. For example, in our small network we have only four of the possible ten links between the five nodes, and the larger the network gets the sparser, in general, it becomes. The mathematical basis of Bayesian networks is now thoroughly developed and they have found numerous applications, for example, in medical diagnosis, information filtering, intelligent user interfaces, and pattern recognition:

Pearl's formulation has had a revolutionary impact on much of AI.

Peter Szolovits and Stephen Pauker (1993), Categorical and probabilistic reasoning in medicine revisited, Artificial Intelligence, 59, 167-180.

The Bayesian network formalism is the single development most responsible for progress in building practical systems capable of handling uncertain information.

Peter Haddawy (1999), An overview of some recent developments in Bayesian problem solving techniques, AI Magazine, 20, 2, 11-19.

So, the Bayesian network formalism is now the core of probabilistic reasoning, as resolution-based theorem proving is the core of logical reasoning.

However, the determination of the extent to which propositions should be believed, on the basis of evidence, is not in itself sufficient to make rational decisions. We also need a theory of utility, which describes the desirability of the possible outcomes. For example, we can reason about the probabilities of various diagnoses but in order to make a decision about the treatment we need to consider factors such as the costs incurred by any operation, delays in treatment, consequent incapacities, and so on. Coupling probabilities and utilities provides a decision theory determining what action should be taken.

Such theories have been developed in economics, particularly: they describe what decisions should be made but not quite what decisions actually are made (by humans, that is). We tend, for example, to not take risks when the probabilities are high, perhaps in order to avoid embarrassment if we are unlucky. Research on the irrational nature of human decision-making by Daniel Kahneman in the 1970s eventually led to his Nobel Prize for economics in 2002. A theory of utility should include such psychological factors if it is intended to model human decision-making.

While Bayesian network techniques are themselves mathematically complex, the representation, which is simply a network of linked propositions, does not have the symbolic status of logic-based methods. Bayesian methods calculate the probabilities of propositions but do not reason with the symbolic content of the propositions. The logic-based and probability-based approaches to plausible reasoning should be seen as complementary and not as rivals. The former helps to determine the propositions that have probabilities worth caring about.

It is not clear what the outcome of the debate about the role of logic in uncertain reasoning will be. At least it needs to be emphasised that although at some level the programming of present computers does require the use of some formal notation this does not necessarily imply that the associated operations need to correspond to the conventional notion of a logical process.

30. Logic programming: "conveniently expressible"

Within AI itself, opinions on the relevance of formal logic have spiralled. Initially, with the notable exception of John McCarthy's Advice Taker proposal, notations used in AI programming were invented ad-hoc to suit the problem. The clearest evidence of the recognition of the potential advantages of using a formal logical notation is seen in one of the earliest AI theses, that of Bertram Raphael, written at MIT in 1964. The first half of the thesis described a program (SIR, for Semantic Information Retrieval) that used an idiosyncratic notation for expressing the meaning of sentences such as "The man owns the car" and "A dog has four legs" and the second half proposed the replacement of this notation by the first-order predicate logic:

> **SIR contains a separate subprogram for determining the 'truth' for each relation in the system … In practice each of the truth-testing subprograms operates by searching the model, looking for certain combinations of attribute links. However, since the existence of an attribute link implies the truth of a corresponding predicate, we may consider the subprogram as deducing the truth of a predicate from the fact that certain other predicates are true. Such deduction procedures are conveniently expressible in the first-order predicate logic.**
>
> *Bertram Raphael (1964), SIR: a computer program for semantic information retrieval, PhD thesis, MIT, Cambridge, Mass. (also in Marvin Minsky, ed. (1968), Semantic Information Processing, Cambridge, Mass.: MIT Press).*

The date (1964) is significant: it precedes the development of the resolution procedure by Alan Robinson in 1965. Before 1965, the suggestion to use predicate logic was only of speculative interest, as there were no practical procedures for reasoning with expressions in predicate logic. But in the years immediately after 1965 there was feverish activity in applying computational logic based on resolution to many problems within AI, such as natural language understanding and robot problem solving.

However, various difficulties such as those mentioned above, in particular, the computational ones, led to a reaction, with Winograd's thesis in the vanguard. The MIT group was especially critical of the reliance on formal logic, with Marvin Minsky's 'frames' paper, published in 1975, being particularly influential. In what may now be seen as an early attempt to develop a theory of context, Minsky proposed a theory of thinking based on the proposal that commonsense inferences may be made by matching a new situation with a previously remembered stereotypical situation:

> **Whenever one encounters a new situation (or makes a substantial change in one's viewpoint), he selects from memory a structure called a frame; a remembered framework to be adapted to fit reality by changing details as necessary … [representing] a stereotyped situation, like being in a certain kind of living room, or going to a child's birthday party.**
>
> *Marvin Minsky (1975), A framework for representing knowledge, in Patrick Winston (ed.), The Psychology of Computer Vision, New York: McGraw-Hill.*

The use of the word 'frame' was unfortunate as the paper had nothing to do with the frame problem, already a key AI concern, and more to do with the 'schema' idea developed by the psychologist Fredrick Bartlett in the 1930s and under contemporary investigation by cognitive psychologists. The term had previously been used by Otto Selz in the 1910s and by Immanuel Kant in his *Critique of Pure Reason* of 1781. Actually, Bartlett disliked the word 'schema' because of the simplistic interpretation that others gave it.

The impetus for the next turn of the spiral already existed – within the thesis of MIT's Terry Winograd! Although arguing against the use of logic, he devised (with Carl Hewitt) a programming language called Planner, which has marked similarities with the language Prolog and the 'logic programming' methodology in general in relying on procedures being invoked when needed rather than when called by name. Prolog is directly based on first-order predicate logic and resolution theorem proving. The genesis of Prolog is a little unclear but it seems to have arisen from attempts to automate the resolution principle, with the first Prolog programs being written by Alain Colmerauer in France in the early 1970s and the major development of the language being carried out by David Warren and colleagues in Scotland later in the decade.

In logic programming one defines a problem with expressions in formal logic and then lets a general-purpose inference mechanism try to prove that a solution to the problem follows. For example, if we are dealing with a problem concerned with who people like and dislike, we could say that everybody likes people who like dogs, their spouse (this is just an example) and themselves, and everybody dislikes politicians and their mother-in-law. In Prolog we might write the following axioms:

```
likes(X, Y):- likes(Y, dogs).
likes(X, Y):- spouse(X, Y).
likes(X, X).
dislikes(X, Y):- politician(Y).
dislikes(X, Y):- spouse(X, Z), mother(Z, Y).
```

The first line, for example, might be read as "X likes Y if Y likes dogs". We might then ask the system `likes(Fred, Y)`, that is, "Who does Fred like?",

or `likes(X, Fiona)`, that is, "Who likes Fiona?", and the system will attempt to find, from its database, who fits the definitions. Clearly, in simple cases like this, Prolog is just a re-expression of predicate logic and the process of answering queries corresponds to applications of the rule of resolution. For example, the first axiom corresponds to

`likes(y, dogs)` → `likes(x, y)`

and given a fact such as `likes(Sue, dogs)` then `likes(x, Sue)` follows from resolution.

In logic programming, the programmer specifies axioms defining predicates relevant to the problem at hand and leaves the system with the task of deriving a solution from them. In other words, programming has a declarative style, in which high-level descriptions of a problem are given, rather than the procedural style of conventional programming, in which a sequence of instructions are specified by the programmer. The logic programming methodology was adopted by Japan in its Fifth Generation Computer Systems project initiated in 1982. One aim of this ambitious project was that:

> **Everyone will be able to converse with computers without a professional knowledge of them, even if everyday language is used, [and] the computers will be able to understand our thoughts and give us suitable answers.**
>
> *Tohru Moto-oka, ed. (1982), Fifth Generation Computer Systems, Amsterdam: North-Holland.*

This omniscience was to be achieved through parallel inference machines that differed from the von Neumann architecture, so much so that their speed was to be measured in logic inferences per second (lips) rather than the more normal million instructions per second (mips).

Today, formal logic seems established as the basis for theoretical AI and is introduced as such in current AI textbooks:

> **[Logic is] the oldest, richest, most fundamental, and best-established branch of mathematics.**
>
> *Michael Wooldridge (2000), Reasoning about Rational Agents, Cambridge, Mass.: MIT Press.*

These, however, do not seem the soundest of reasons for adopting formal logic, when we are looking for the most appropriate, relevant, useful and efficient form of representation for a rational computer system, which:

> **... chooses to perform actions that are in its own best interests, given the beliefs it has about the world.**
>
> *Michael Wooldridge (2000), Reasoning about Rational Agents, Cambridge, Mass.: MIT Press.*

As we have seen, the best choices are not necessarily those that a deductive logical system would recommend.

The syntax of logic forces writers to be precise in what they are saying, which is not always the case when idiosyncratic notations are invented. The semantics provide a universally understood way of assigning meanings to the symbols, which, again, is often far from the case with other notations. The process of drawing conclusions from logical representations is computationally well defined and is perhaps as efficient as it can be. The process bears comparison with human reasoning processes, although logical inferences are deductive, whereas much human reasoning is not. Some plausible reasoning methods can be expressed in deduction-like terms but processes of induction (that is, forming general rules from specific cases) and abduction (that is, forming explanations for observations) seem substantially different. Logicians have yet to formalise the process of seduction, in which propositions that would normally be considered untenable somehow become acceptable.

At least it is clear that the representation of declarative knowledge demands a notation that is as expressive as first-order predicate logic at the minimum. Standard logic provides a basis for analysing processes of reasoning and perhaps can be extended to address aspects of reasoning that are not well handled by standard logic. It would seem misguided to neglect the theoretical apparatus that has already been developed in the hope that some new, alternative approach can somehow overcome the limitations of logic without introducing further problems.

31. Knowledge: "is power"

Arguments about the use of predicate logic within AI are concerned mainly with the *form* of knowledge representation. There is a related argument concerned with the *content* of knowledge representation. Early AI placed the emphasis on knowledge-independent problem solving techniques (as in GPS) and did not concern itself too much with the specific knowledge which programs actually needed to solve problems. Logic-based methods also separated process from content, with a focus on the former in order to develop efficient computational techniques. During the 1970s, however, it became increasingly argued that ignorance was no longer bliss. General techniques were considered too weak to provide powerful problem solving performance – more important was the specialist knowledge that needs to be brought to bear.

Advocates of this point of view tended to quote the slogan:

Knowledge is power [the usual translation of **nam et ipsa scientia potestas est].**

Francis Bacon (1597), de Haeresibus, Meditationes Sacrae.

However, Bacon (who is considered the founder of the 'scientific method' although he made no major discoveries and revealed no new laws himself) was really commenting that knowledge provided power or control over nature, not promising that knowledge would assure political, commercial or industrial power, as the advocates hoped to imply. Today, the implication has been fully absorbed:

Knowledge in the form of an informational commodity indispensable to productive power is already, and will continue to be, a major – perhaps *the* major – stake in the world-wide competition for power.

Jean-François Lyotard (1979), The Postmodern Condition: A Report on Knowledge.

Douglas Lenat and Edward Feigenbaum adopted and adapted the slogan to form what they called the 'knowledge principle', intended to encapsulate the wisdom gained after a decade or so of research and development on 'knowledge-based systems':

… we can summarize the empirical evidence: "Knowledge is Power" or, more cynically, "Intelligence is in the eye of the (uninformed) beholder". The *knowledge as power* hypothesis has received so much confirmation that we now assert it as:

Knowledge Principle: A system exhibits intelligent understanding and action at a high level of competence primarily because of the *knowledge* that it can bring to bear: the concepts, facts, representations, methods, metaphors, and heuristics about its domain of endeavor.

Douglas Lenat and Edward Feigenbaum (1991), On the thresholds of knowledge, Artificial Intelligence, 47, 185-250.

The claim is that systems gain competence by being provided with specific knowledge enabling them to get quickly to the heart of the problem, without having to flounder around applying general methods during an exhaustive search.

As we have already seen, AIers are particularly disputatious and inevitably this emphasis on specialist knowledge was accompanied by claims that it represented a paradigm change from the first decade or so of AI, which was characterised as having failed because it had been concerned only with general-purpose problem solving methods:

The first period of AI research was dominated by a naive belief that a few laws of reasoning coupled with powerful computers would produce expert … performance. As experience accrued, the severely limited power of

> **general-purpose problem solving strategies ultimately led to the view that they were too weak to solve most complex problems.**
>
> *Frederick Hayes-Roth, Donald Waterman and Douglas Lenat, eds. (1983), Building Expert Systems, Reading, Mass.: Addison-Wesley.*

In fact, the development of knowledge-based systems was very much in the symbol-processing tradition of AI and was a change of focus, not of paradigm. Considering general-purpose methods and specialist knowledge to be in opposition is nonsensical because clearly both are needed, as AI work itself shows. The general methods achieve nothing without knowledge to work with and, as we will see, the methods used in knowledge-based systems are virtually identical to the vanquished general-purpose methods. The fanfare with which knowledge-based systems were launched owed more to the need to disassociate them from previous AI work, which was perceived as making slow progress, and to establish a market presence, for it was considered necessary to apply AI to real-world problems not to the toy problems considered up to then.

The thaumaturgic ingredient in these new AI products was to be specialist knowledge, narrowing the central role that the computer had already taken in the information society:

> **As the central organizer and repository of information – the basis of knowledge – the computer is called upon more and more to play a massive role in this new society. Knowledge, rather than capital, labour, or raw materials, has become the major source of economic growth.**
>
> *Sam Wyly (1972), As the industry sees it, Communications of the ACM, 15, 516-517.*

And so was born the 'knowledge industry', an industry based on the idea that the most important possession that a company has is its knowledge. Of course, it may be a category mistake to say that a company possesses knowledge: if knowledge is something that can be possessed by anything, it is by the community of employees of that company. Unfortunately, employees are unreliable – they may pass away, forget, or take their knowledge elsewhere. Also, if their knowledge is worth having, they are expensive. It therefore makes business sense for companies to protect their most valued possession, by ensuring that knowledge does not belong only to transient employees.

This may have seemed a revolutionary idea to AIers but others saw it as a culmination of the 'Taylorization' of the workforce, which began in the early twentieth century with the principles of 'scientific management' of Frederick Winslow Taylor (1856-1915). At that time, productivity was limited by the fact that the factory workers, who had the craft skills and were

paid at piece rates, had no incentive to increase output because if they did the piece rate would be reduced and they would have to work harder for the same income. Managers had little idea of the working practices and so did not know what output rates were reasonable. To overcome the problem, Taylor proposed that management gather up and systematize the craft knowledge that the workforce had acquired:

> **The managers assume … the burden of gathering together all of the traditional knowledge which in the past has been possessed by the workmen and then of classifying, tabulating, and reducing this knowledge to rules, laws, and formulae which are immensely helpful to the workmen in doing their daily work.**
>
> *Frederick Winslow Taylor (1911), The Principles of Scientific Management, reprinted in 1967, New York: Norton.*

The knowledge-based systems industry proposed to go further, in making system designers and computers, rather than managers, responsible for this task. While so-called scientific management often did lead to increased productivity and higher living standards, its faith in technocratic top-down planning and its patronisation of the workforce eventually came to be considered a malign influence on industrial practice. Having passed knowledge of their skills to the management, workers were then expected to just follow instructions. Of course, knowledge-based systems would not make those mistakes: they were intended to systematize specialist knowledge, not the knowledge of run-of-the-mill workers.

With knowledge-based systems, the focus is on specialist knowledge not on the commonsense knowledge that was supposed to overcome the limitations of straightforward reasoning mechanisms:

> **"Sometimes you don't use your common sense, Dick."**
>
> **"I never understood what common sense meant applied to complicated problems – unless it means that a general practitioner can perform a better operation than a specialist."**
>
> *F. Scott Fitzgerald (1934), Tender is the Night, New York: Charles Scribners' Sons.*

Whether the knowledge is that of everybody or just a special few, we need first to consider how knowledge-based systems are to deal with it. Following the AI practices of the time, it was assumed that to provide a machine artifact with knowledge it was necessary to represent that knowledge in symbolic form.

In general, though, philosophers wince at AI's cavalier use of the term 'knowledge', as they have themselves filled the shelves of libraries trying to define what it is. A starting point for these discussions is often a definition such as 'knowledge is justified, true belief', a point which actually started

some considerable time ago, being traceable to Plato's *Theaetetus*. 'Belief' is therefore taken as a basic concept, qualified in two ways to constitute knowledge. I may believe that it is snowing in England but I cannot reasonably say that I know it unless it is in fact the case that it is snowing in England and that I have a justification for believing so, such as, for example, that I have just seen a live television broadcast from England:

> **'England,' said Christophine, who was watching me. 'You think there is such a place?'**
>
> **'How can you ask that? You know there is.'**
>
> **'I never see the damn place, how I know?'**
>
> **'You do not believe that there is a country called England?'**
>
> **She blinked and answered quickly, 'I don't say I don't *believe*, I say I don't *know*, I know what I see with my eyes and I never see it.'**
>
> *Jean Rhys (1966), Wide Sargasso Sea, London: André Deutsch.*

So, for some, seeing is more than believing, a view of some philosophical pedigree, being part of the doctrine of phenomenalism, which holds that human knowledge is confined to the appearances presented to the senses.

Other kinds of justification are, of course, possible, and philosophers will debate their acceptability, but we need pursue this no further than to remark that this is a very 'static' notion of knowledge. It is one that encourages us to write something like `Weather(England, snowing)` to represent the belief or knowledge, as though that is all there is to it. The venerable traditions of epistemology, that is, the theory of knowledge, have been concerned with questions such as "How, if at all, can we ever know anything?" and "Do we gain knowledge through our senses or through reasoning or both or neither?" and while these should, of course, be of interest to AIers they tend to be more interested in the relationships between knowledge and action, that is, with questions such as "How can a robot know when it has enough knowledge to perform some action?" and "How can a system determine how an action would affect what it already knows?" In AI the computational representation of knowledge is not a philosophical exercise – it is a means to an end, the end being to solve problems effectively:

> **Action is the proper fruit of knowledge.**
>
> *Thomas Fuller (1732), Gnomologia: Adages and Proverbs, 760.*

The applied purpose of knowledge has long been acknowledged, along with what to do in its absence:

> **The essence of knowledge is, having it, to apply it; not having it, to confess your ignorance.**
>
> *Confucius (551?-479 BC).*

Confessing ignorance, however, does not come easily to AIers. The zeal of knowledge-based crusaders scarcely acknowledged that we could not expect to represent knowledge that we do not have, which is a significant limitation in general and particularly for specialist knowledge, which, by definition, is not known by ordinary mortals, such as system designers:

Our knowledge can only be finite, while our ignorance must necessarily be infinite.

Karl Popper (1963), Conjectures and Refutations: The Growth of Scientific Knowledge, London: Routledge & Kegan Paul.

There is no absolute knowledge. And those who claim it, whether they are scientists or dogmatists, open the door to tragedy.

Jacob Bronowski (1973), The Ascent of Man, Boston, Mass.: Little Brown & Co.

32. Expertise: "very special and narrow"

The idea of a knowledge-based system is that it be able to solve complex problems by recourse to a computational representation of the specialist knowledge needed to address such problems. Typically, knowledge-based systems tackle problems which otherwise would be presented to human consultants, such as problems in medical diagnosis, financial management, or the law. The human consultants would have acquired their own knowledge from many years of experience in addition to academic study, and it is the knowledge-based system designer's task to transcribe this experiential knowledge into computational form. As a token of the reverential esteem in which we hold such consultants, the corresponding knowledge-based systems are often called 'expert systems'.

What exactly is an 'expert'?:

An expert is someone who knows some of the worst mistakes that can be made in his subject and how to avoid them.

Werner Heisenberg (1971), Physics and Beyond, New York: Harper and Row.

An expert is a man who has made all the mistakes, which can be made, in a very narrow field.

Niels Bohr, quoted in Alan Mackay (1977), The Harvest of a Quiet Eye: a Selection of Scientific Quotations.

An expert is a person who avoids the small errors while sweeping on to the grand fallacy.

Arthur Bloch (1979), Murphy's Law, Los Angeles: Stern Sloan Publishing.

Experts invent themselves. Whereas I was born with my mind made up.

J.L. Carr (1975), How Steeple Sinderby Wanderers Won the F.A. Cup, London: Prior Books.

> **The essence of the expert is that his field shall be very special and narrow: one of the ways in which he inspires confidence is to rigidly limit himself to the little toe; he would scarcely venture an off-the-record opinion on an infected little finger.**
>
> *Louis Kronenberger (1954), Company Manners: A Cultural Inquiry into American Life, Indianapolis: Bobbs-Merrill.*

A 'little toe expert' system or a little toe 'expert system' is a very different kind of computer program from GPS. Superficially, the latter knew nothing except a few general techniques, whereas the former knows everything about little toes and nothing about anything else. There is, of course, an empirical question as to whether it is in fact possible to know everything about little toes without a great deal of less specialist knowledge to support it, for if not, the latter would presumably also need to be accessible to our little toe expert system.

In fact, expert systems are renowned for their brittleness. A slight deviation from the system's area of competence may lead to large and usually calamitous changes in performance. To overcome this problem, designers have attempted to develop ontologies that specify precisely the boundaries of a system's expertise, or have tried to enable systems to adapt dynamically (that is, to learn) from feedback from the environment. While this brittleness is regarded as a shortcoming of expert systems, it is perhaps preferable to an expert on one topic (say, the book of Genesis) being automatically considered qualified to offer opinions on another (say, genetic engineering).

Many studies have been carried out comparing human expert and non-expert performance in order to try to pin down the nature of expertise. It seems to have something to do with the following factors: experts just know more stuff than non-experts; experts organise and use their knowledge better; experts analyse problems better; experts are more skilled, that is, they do things more adeptly than non-experts; experts have more practical experience (in some fields, they may have practical experience but no professional qualification); experts are more creative; and so on. These may seem platitudinous or definitional. As far as expert systems are concerned, the focus has mainly been on the quantity of knowledge needed for expertise.

Although we should be wary of placing our trust in expertise, whether human or computational – for example, the distinguished philosopher Karl Popper would like to save us ...

> **... from narrow specialization and from an obscurantist faith in the expert's special skill, and in his personal knowledge and authority; a faith that so well fits our 'post-rationalist' and 'post-critical' age, proudly**

dedicated to the destruction of the tradition of rational philosophy, and of rational thought itself.

Karl Popper (1934), Logik der Forschung, published in English as The Logic of Scientific Discovery (1959), Hutchinson: London.

we might, nevertheless, decide to try to build an expert system. The first thing we might think to do is to read the textbooks on the subject matter. However:

The most important observations and turns of skill in all sorts of trades and professions are as yet unwritten. This fact is proved by experience when passing from theory to practice we desire to accomplish something.

Gottfried Wilhelm Leibniz (1646-1716), in P. Wiener, ed. (1951), Selections, New York: Scribner.

What was true in Leibniz's time is still true today. What we are trying to computationalise does not exist in books.

The next thing might be to talk to human experts to find out what knowledge they use to solve problems:

Experience has also taught us that much of this knowledge is private to the expert, not because he is unwilling to share publicly how he performs, but because he is unable.

Edward Feigenbaum (1977), The art of artificial intelligence, Proceedings of the 5th International Joint Conference on Artificial Intelligence, 1014-1029.

Many people can talk sense with concepts but cannot talk sense about them; they know by practice how to operate with concepts, anyhow inside their chosen fields, but they cannot state the logical regulations governing their use. They are like people who know their way about their own parish, but cannot construct or read a map of it, much less of the region or continent in which their parish lies.

Gilbert Ryle (1949), The Concept of Mind, New York: Barnes and Noble.

In other words, the knowledge that experts have, or appear to have since they are able to solve problems that seem to require it, is knowledge which may be acquired neither from reading text-books, since it is not there, nor by listening to experts, since they are unable to verbalise their own knowledge:

All craftsmen share a knowledge. They have held reality down fluttering to a bench.

Vita Sackville-West (1927), The Land, 'Summer'.

The prospects, then, for capturing this knowledge to represent within expert systems appear to be bleak.

However, expert systems designers developed techniques that had sufficient success to enable them to call themselves 'knowledge engineers'. The analogy with mining engineering permeates expert system literature,

partly to show that this is a real, sleeves-rolled-up kind of activity, not the airy-fairy sort of AI we had heretofore. As with any analogy, there are some features that we are supposed to know apply to both mining engineering and knowledge engineering; some we are supposed to know do not apply to both; and some we are not sure about. Knowledge is a resource; it is valuable; it is rare; it is buried and needs to be brought to the surface; it can be refined; it needs a qualified engineer to get at it.

But despite being 'extracted', knowledge stays where it was; once extracted, knowledge can be infinitely duplicated; if not extracted, knowledge does not stay where it was forever; the 'site' of extraction (the human expert) is not passive in the extraction process. But does knowledge exist in isolated 'nuggets' that can be extracted one by one? Can it be transported and given to others (humans or systems)?

If the expert's private knowledge can indeed be extracted then:

Building expert systems is a form of intellectual cloning.

Randall Davis (1984), Amplifying expertise with expert systems, in Patrick Winston and Karen Prendergast (eds.), The AI Business, Cambridge, Mass.: MIT Press.

In this case, an obvious question arises: why would human experts, with expertise that they can see is so valuable that it is considered worthwhile trying to extract it, go along with this process, thereby rendering themselves redundant? The answer usually given by knowledge engineers is the rather self-contradictory one that the expert system will only handle the routine problems that are a waste of the human expert's valuable time, leaving him or her to focus on the challenging and important problems where expertise is really needed and for which the expert is truly valued. One thing is clear: unlike mining, knowledge engineering is or should be a collaborative process, involving a constructive interaction between the engineer and the possessor of the resource, the expert.

Despite this notion of cloning and the attempts to acquire system knowledge from the only available source, that is, human experts, expert system designers emphasise that their systems are results-oriented: the systems are intended to solve practically important problems effectively and efficiently. There is no interest in whether the systems operate in ways analogous to humans:

These efforts are not concerned with similarities between resulting systems and human performance ... They are intended simply to perform the task without errors of any sort, humanlike or otherwise.

Bruce Buchanan and Edward Shortliffe (1984), Rule-based Expert Systems, Reading, Mass.: Addison-Wesley.

Bruce Buchanan and Edward Shortliffe were leaders of the two earliest large-scale expert system projects, those of DENDRAL (1965-79 or so) and MYCIN (1972-84 or so). Therefore, they have exemplary qualifications to comment that expert systems are not concerned with similarities to human performance, and yet this remains one of the most confusing aspects of expert systems work. Although claimed to be the most applied part of the AI field, interested only in developing systems that are practically successful, expert systems are also the ones that rely most on understanding how humans solve the problems that the systems address. This seems unavoidable because designers do not themselves have the specialist knowledge that is needed for the expert systems. However, we have the case history of chess playing programs to show that it is possible to develop expert-level performance without needing to study human experts in detail. Nonetheless, overall it is disingenuous to say that expert systems are not intended to work in similar ways to humans solving the same problems.

33. Rule-based systems: "logical basis is obscure"

The first system to demonstrate the importance of specialist knowledge was the DENDRAL system mentioned above, which was concerned with identifying chemical structures from nuclear magnetic resonance readings. However, it was not implemented in the style that became standard for expert systems. This style was first convincingly illustrated by MYCIN, which aimed to diagnose blood infections. The confusion about the psychological status of expert systems is compounded by the unfortunate coincidence that the standard computational representation for the knowledge in an expert system is as a set of rules, called a production system, of the same syntactic form as Newell and Simon and others had developed for their cognitive models:

> **We confess to a strong premonition that the actual organization of human programs closely resembles the production system organization.**
>
> *Allen Newell and Herbert Simon (1972), Human Problem Solving, Englewood Cliffs, N.J.: Prentice-Hall.*

A priori, it seems unlikely that a notation considered suitable for expressing human cognitive processes, complete with all its limitations and fallibilities, is also ideal for the efficient computation required of expert systems. Unless, that is, designers and theorists feel that there is something fundamental about the 'rule', as a generalisation of the stimulus-response connections of earlier psychological theories, in a decision-making process.

In a production system, each rule is of the form:

```
If condition1 and condition2 and ...
   then conclusion1 and conclusion2 and ...
```

For example, a rule in a simple medical expert system might be:

```
If there is a tingling sensation in the hands
and there is a stiffness in the neck
   then cervical spondylosis may be the cause.
```

Usually, the rule does not lead to an immediate solution to the problem but provides an intermediate conclusion that may be used as a condition in some other rule.

A typical production system has three parts. First, we have the rules themselves, each consisting of a set of conditions and a set of conclusions or actions. The conditions determine whether the rule is applicable, that is, whether the conclusions or actions should follow. The second part is a description of the current problem solving state, which is called the 'working memory' in psychologically oriented applications. The conditions are matched against this description. The actions, if executed, change the description of the problem solving state. The third part is the 'recognise-act cycle', which repeatedly matches conditions to find applicable rules and then selects one of these rules for which the actions are then carried out. Various schemes are used to select the rule to activate, for example, the most specific applicable rule or the one added to the set of rules most recently. The cycle continues until the problem is solved or no rules are applicable.

Since the expert finds it difficult to talk about his knowledge in the abstract, the knowledge engineer typically builds a prototype system with only a few simple rules and this serves as a catalyst for the expert to criticise. In this way, the set of rules is slowly expanded and refined:

This private knowledge can be uncovered by the careful, painstaking analysis of a second party, or sometimes by the expert himself, operating in the context of a large number of highly specific performance problems.

Edward Feigenbaum (1977), The art of artificial intelligence, Proceedings of the 5th International Joint Conference on Artificial Intelligence, 1014-1029.

The errors of a wise man make your rule rather than the perfections of a fool.

William Blake (1811), English Encouragement of Art: Cromack's Opinions put into Rhyme.

Any fool can make a rule and every fool will mind it.

Henry David Thoreau (1860), Journal, February 3.

The outcome, if the process is successful, is a set of hundreds or thousands of specific rules, laboriously extracted by the knowledge engineer from the expert or experts over perhaps a year or more.

Indeed there are examples of successful expert system projects. One of the first was a system called XCON (originally R1 but renamed to avoid having to keep repeating a feeble joke) developed by the company DEC to determine the configuration, that is, the components, cabling, and so on, for an ordered computer system. Another, called Prospector, recommended where to drill for minerals and entered expert system mythology for a successful prediction of a molybdenum deposit, although its designers later clarified the rather limited extent of Prospector's contribution.

A production system interpreter may be data-driven or goal-driven. We may take the data and see what conclusions follow, reasoning from left to right of the rules. Or we may take the goal and regard it as a desired conclusion, matching it with the right-hand-side of rules to find one that is relevant and then see if the conditions (on the left-hand-side) are true. Any condition that cannot be immediately shown to be true then becomes a further conclusion to derive, and so on, recursively. This process sounds very similar to that of deriving a conclusion by the repeated application of a rule of inference, such as resolution, to a set of premises. In fact, they are formally equivalent and therefore all the difficulties that arise with logical reasoning also arise with reasoning with production systems. However:

Although production rules are widely used in AI, they frequently lead to ad hoc systems whose logical basis is obscure.

John Sowa (1984), Conceptual Structures: Information Processing in Mind and Machine, Reading, Mass.: Addison-Wesley.

This is because the syntax and semantics of the rules of production systems do not have a standard definition, as predicate logic does.

A production system interpreter is, therefore, a general-purpose problem solving method, which knowledge engineers had disdained. It performs in exactly the same way, matching conditions, performing actions, and so on, regardless of the specialist knowledge that it is working on. This is made very clear by the fact that the interpreter can be isolated as an 'expert system shell', to be filled with different specialist knowledge to create different expert systems: at least, that is what its marketers claimed.

Saying that a production system interpreter works much like a resolution theorem-prover raises the question of how production systems overcome the limitations discussed for the latter. In particular, we cannot expect the system to provide certifiably correct solutions because, for one thing, production rules may not be 100% reliable and neither may the data itself. For example, the rule above indicates that "cervical spondylosis may be the problem", which might mean that the diagnosis is correct in, say, 70% of cases. Early expert systems such as MYCIN and Prospector had ad-hoc procedures for

reasoning with such probabilities although they have now been displaced by more rigorous schemes such as Bayesian networks.

Newell and Simon's claim that 'human programs' are like production systems is a claim about the organisation of the human mind. It is not a claim about what the mind may compute, because we know that in a formal sense all computations are equivalent. Production systems may, of course, be implemented in Lisp or any other programming language, but they operate in a different style to normal Lisp programs. Typically, Lisp functions are written and called by name, as specified by the programmer, but in a production system there are a large number of programs (or rules) that are activated only when the situation is appropriate. In this sense, production systems are closer to the declarative style of logic programming.

34. Knowledge engineering: "increaseth sorrow"

Returning to expert systems, which are, of course, application-oriented, it was found that some components, for example, to generate explanations, to perform meta-reasoning, and to acquire new rules, needed further research. Because there is no assurance that a knowledge base is complete or sound, expert systems forego any pretence at providing output that is guaranteed to be correct. Expert systems tackle problems for which it is usually possible to offer only plausible solutions and hence it is necessary to provide some kind of explanations of their solutions, which conventional software is unable to do. In fact, the increasing complexity of software systems has led to a concern that all systems should include a capability for providing explanations, so that users of the systems may be able to form some judgement about the worth of the answers provided.

The basic idea of explanation generation had been indicated by Winograd's SHRDLU:

User: **When did you pick up the green pyramid?**
SHRDLU: **While I was stacking up the red cube, a large red block, and a large green cube.**
User: **Why?**
SHRDLU: **To get rid of it.**
User: **Why?**
SHRDLU: **To clean off the red cube.** (and so on, until)
User: **Why?**
SHRDLU: **Because you asked me to.**

Terry Winograd (1972), Understanding Natural Language, New York: Academic Press.

The technique for generating such explanations is to record the steps through which the answer was derived and then report them back to the user: "Why do you think it is cervical spondylosis?" "Because there is a tingling sensation in the hands." And so on.

Unfortunately, it is not always that simple. There are generally so many steps that to list them all would lead to ridiculously verbose and hence incomprehensible explanations. The steps have to be abstracted in some way and a coherent summary of them created:

A little inaccuracy sometimes saves tons of explanation.

Saki (H.H. Munro) (1924), The Square Egg, London: Bodley Head.

We might imagine that the explanation should emphasise only the important steps. However, what is judged important depends upon the interests and knowledge of the person receiving the explanation. We can also anticipate that the terms used within the program to define the knowledge used to derive answers may not correspond to the concepts with which the user is familiar. Perhaps we need to embark on a philosophical investigation of just what constitutes an 'explanation':

I have an explanation for all this but I am not sure that I believe it myself.

Groucho Marx.

My advice to you is not to inquire why or whither, but just enjoy your ice-cream while it's on your plate.

Thornton Wilder (1942), The Skin of our Teeth, Act 1.

With that advice, let us turn to the issue of meta-reasoning.

A problem that arises with knowledge bases of thousands of rules is that of efficiently finding those rules that are applicable. This is formally exactly the same problem as that of determining the most useful premises to derive a desired conclusion, from the large number of potentially relevant premises available.

In the case of knowledge-based systems, some optimism was held for the use of meta-knowledge, that is, rules which express knowledge about the knowledge base. A statement such as "I know nothing about art" might be considered to be an expression of meta-knowledge: it is a statement about what I know or believe about my own knowledge. Similarly, we might imagine that it would be useful for a knowledge-based system to represent, for example, that the knowledge base contains no rules concerned with art, so that the system wastes no time trying to derive a conclusion about art. In fact, we might view an explanation, as above, as a form of meta-knowledge, since it involves an abstraction of the knowledge base, rather than an execution of it.

Some philosophically oriented researchers are particularly intrigued by the notion of meta-knowledge, considering that it provides a means to develop systems with reflective self-awareness, which some people regard as a uniquely human aspect of consciousness:

Knowing others is wisdom; knowing the self is enlightenment.

Lao Tse (604-531 BC).

Full wise is he that can himselven knowe.

Geoffrey Chaucer (1387), The Monk's Tale, The Canterbury Tales.

Our thesis is that it is more rewarding for long range applications to understand what are the basic concepts needed to implement powerful meta-descriptions, rather than building plethoras of "expert" production systems.

Gerard Huet (1982), In defense of programming languages design, Proceedings of the European Conference on Artificial Intelligence.

The ability to reason at a meta-level about the contents of a knowledge base is greatly facilitated by the fact that the individual rules are, in principle, independent of one another. Ideally, each extracted rule in a production system represents a separate nugget of expertise. It should therefore be understandable and explainable in isolation.

If each rule *is* independently comprehensible, it is much easier to modify the rules and to add new ones, since there is no need to worry about interactions with other rules. In contrast to conventional software, expert systems are incomplete in the sense that it was always possible to change or add rules. It was envisaged that the knowledge acquisition process would proceed through the knowledge engineer and the expert discussing the prototype system and incrementally revising and accumulating rules.

This prospect led to a ferment of activity in which many industrial companies invested considerable resources in the expectation of an eventual pay-off different to that predicted by an unimpeachable source:

Many shall run to and fro, and knowledge shall be increased.

Bible, Daniel 12, 4.

He that increaseth knowledge increaseth sorrow.

Bible, Ecclesiastes 1, 18.

As was mentioned previously, the really useful knowledge for expert systems could not be found in textbooks or by directly asking experts. It was necessary to engage experts in a lengthy criticism of computer-generated examples of the 'knowledge-to-be in action'. No doubt this carries philosophical implications about the nature of this kind of knowledge – on the relationship between implicit and explicit knowledge and the means of converting one to the other through critical interaction, and on the way

knowledge can only be addressed by actively situating it within a problem-solving context in the world:

> **The knowledge of the world is only to be acquired in the world, and not in a closet.**
>
> *Lord Chesterfield (October 4 1746), Letters to his Son.*
>
> **If you want knowledge, you must take part in the practice of changing reality. If you want to know the taste of a pear, you must change the pear by eating it yourself.**
>
> *Mao Tse-Tung (1966), Quotations from Chairman Mao Tse-Tung, Chapter 22.*

Mao Tse-Tung is here expressing the notion of 'grounding', that is, the philosophical view that knowledge has to be grounded in sensory experiences of the world. According to this view, it is not enough to create a knowledge base with an entry such as `Taste(pear, sweet)`: any such expression ought to have arisen from the direct experience of tasting pears. If this view is valid, then very few expert systems can be said to know anything because they cannot experience real medical diagnosis, mineral exploration, and so on. Knowledge, or a substitute for it, has to be gained from the mediation of human experts.

The outcome of this knowledge engineering effort is not only that some profitable expert systems have been implemented but also that the pool of knowledge in the world has grown. Whereas previously the knowledge of how to perform certain skills (such as the determination of the molecular structure of chemical compounds from mass spectrographs) existed only with a few human experts, it now exists in an explicit form within computer programs, that is, in a form that may be inspected, modified and studied. Perhaps more importantly, we have discovered a new way to represent knowledge, that is, as executable programs. With this comes the realisation that if knowledge is to be of any use to its possessor then this may well be the most natural representation, in principle. For example, knowledge of symbolic integration has previously been represented in textbooks in a 'static' form involving equations, formulae, theorems, tables, and so on, none of which actually describes and shows us how to do symbolic integration. A computer program to perform symbolic integration contains the knowledge in a 'dynamic' form, which could, in principle (since there is much work to do to make computer programs easily understandable by average students), be studied to understand how to do integration.

Unfortunately, the acquisition of new rules for expert systems proved so difficult that it gained its own sobriquet: the 'knowledge acquisition bottleneck'. Considering the many person-years that had been devoted to the earliest expert systems, it perhaps should not have been a surprise that the

efficient development of expert systems foundered on the difficulty of developing the computational representations of the specialist knowledge that our desired expert systems needed:

> **It is a great nuisance that knowledge can only be acquired by hard work. It would be fine if we could swallow the powder of profitable information made palatable by the jam of fiction.**
>
> *W. Somerset Maugham (1954), Novels and Their Authors, Oxford: Heinemann.*

But AIers are nothing if not optimistic. The incremental development of production rules should, they thought, have been straightforward and so, if we were having difficulties with cussedly unhelpful human experts, they would try to bypass them by developing expert systems that learn the rules by themselves from data presented to them or discovered by the systems. Since it is usually considered that humans need at least ten years of dedicated study and extensive practical experience to reach a level acknowledged as expertise, this may seem ambitious. Let us therefore consider the general problem of learning first.

35. Learning: "a prerequisite"

For some, the process of learning is the quintessence of intelligence. Having a program that could solve a problem, however complex, would not constitute a demonstration of intelligence unless the program was itself able to *learn* how to solve that problem:

> **The ability to learn, to adapt, to modify behavior is an inalienable component of human intelligence. How can we build truly artificially intelligent machines that are not capable of self-improvement? Can an expert system be labeled 'intelligent', any more than the Encyclopedia Britannica be labeled intelligent, merely because it contains useful knowledge in quantity? An underlying conviction of many ML [machine learning] researchers is that learning is a prerequisite to any form of true intelligence.**
>
> *Jaime Carbonell (1989), Paradigms for machine learning, Artificial Intelligence, 40, 1-9.*

Learning, then, is a necessary feature of intelligence but not a sufficient one (because all animals learn to some extent but we would consider few of them to be intelligent).

As always, it would help to have a precise definition but learning is hard to pin down. A standard psychology textbook on learning offers:

Learning refers to the change in a subject's behavior or behavior potential to a given situation brought about by the subject's repeated experiences in [that] situation.

Gordon Bower and Ernest Hilgard (1981), Theories of Learning, New York: Prentice-Hall.

This definition does not say that the change has to be for the better but in a computational context that would certainly be the intention. It also does not allow learning through reflection alone but only through experience – and repeated experience at that – in the world.

As we noted earlier when considering the stored program concept, once it was realised that operations could be coded in the same way as operands and hence could be modified like operands, it was understood that we have a basis for writing self-modifying programs, that is, programs that could change their own operations and hence perform differently over time, which is presumably one manifestation of learning:

The instructions being coded in numerical form, there is no longer any distinction between these and the numerical values with which the problem is concerned. Modifications to the instructions can be calculated and made automatically in the course of the computation.

IBM (January 27 1948), brochure for SSEC.

Some of the earliest AI programs were in fact learning programs. Generally, the mechanisms did not involve the modification of internal operations, since that might inadvertently lead a program to run amok, but the isolation of parts of the decision-making process for the specific purpose of enabling the program to change them.

For example, a program to learn to play checkers (or draughts) decided on its moves after evaluating possible board positions by calculating various features of a position, such as the ratio of black to white pieces, the number of isolated pieces, and so on, and then multiplying each value by a 'weight', that is, a number intended to represent the importance of that feature, to come up with an overall evaluation of that board position. The program designer, Arthur Samuel, did not know in advance definitive values for the weights and therefore arranged for the program progressively to adjust the weights as it played games of checkers and thereby learn to make better evaluations:

Wearing all that weight of learning lightly like a flower.

Alfred Lord Tennyson (1809), Conclusion, Stanza 10.

Actually, although Samuel's program is often described as learning from repeated experiences at playing checkers, as Bower and Hilgard's definition required, it did not do so at all. It learned as follows. Imagine that a particular board position, one that you might reach by making a single move from the

current position, has been evaluated to give a value, say, 23 (where the higher the value, the better the position is for you). Imagine that we now evaluate all the positions that your opponent might reach in response to your move, giving values of, say, 15, 25, 26 and 18. Your opponent would select the best position from his point of view, that is, the one with value 15. So 15 would have been a better estimate than 23 of the value of the previous board position. The evaluation is only an approximation but if it were perfect then these two numbers would be the same (because a perfect evaluation would take account of all possible responses). In the present case, the initial evaluation is too high. We therefore tweak the weights, decreasing the positive ones and increasing the negative ones, to make it lower. We would, of course, get more reliable values if we looked more moves ahead. All this is happening regardless of whether the program wins or loses. It doesn't really need to happen within the context of a game.

Eventually, after many thousands of (simulated) games, the program's weights had changed sufficiently for it to become a much better checkers player than Samuel himself, an outcome that again questions the simplistic view that programs can only do what their programmers have programmed them to do and are therefore only as intelligent as their programmer.

The same fundamental idea was being investigated at the same time (the late 1950s) by hardware-oriented researchers. They built machines called perceptrons that contained very large numbers of simple devices, corresponding to Samuel's features. These devices just detected, for example, whether or not a particular point on a drawing was black or white, giving a value of 1 or 0 respectively, and then multiplied all these values by weights in order to determine, on the basis of whether the final overall value was greater than some threshold value or not, whether the drawing was, for example, a particular letter, such as P. The idea was that by showing the perceptron thousands of examples of Ps and non-Ps, and by adjusting the weights according to whether the machine correctly identified a P or non-P, the perceptron would eventually learn to discriminate Ps from non-Ps, that is, it would have learned the concept of 'P-ness'.

At the time, there was considerable excitement about the prospect of such devices eventually learning something really worthwhile, and indeed the experimental studies continued to impress observers for some time:

> **Experiments by ... Rosenblatt and others demonstrate that machines can learn from their mistakes, and in certain limited kinds of learning, outstrip human students.**
>
> *Alvin Toffler (1970), Future Shock, London: Bodley Head*

The kind of learning possible with such devices certainly was limited. It is now generally said that Marvin Minsky and Seymour Papert ended this line of research with their book *Perceptrons: An Introduction to Computational Geometry* (1969), which gave a theoretical analysis showing that a perceptron is inherently incapable of learning apparently simple concepts. For example, it cannot learn the concept of 'connectedness', that is, it would never learn to be able reliably to say 'yes' if by following adjacent black points in a line drawing it is possible to reach all other black points and 'no' otherwise. In fact, their analysis applied to only an impoverished form of perceptron and it is more likely that its timing provided an excuse rather than a reason to curtail funding for a research programme that did not seem to be progressing:

I have to conclude (and here I don't think I am in the minority) that this line of research didn't get anywhere. The discovery task was just so horrendous for those systems that they never learned anything that people didn't already know.

Herbert Simon (1983), Why should machines learn?, in Ryszard Michalski, Jaime Carbonell and Tom Mitchell (eds.), Machine Learning, Palo Alto: Tioga.

36. Symbolic learning: "largely explanation driven"

Researchers on perceptron-based learning hoped that general-purpose learning might be achieved with little or no initial structure or knowledge. Somehow, incremental improvement, by means of successive modifications of numerical parameters, is expected to materialise through the somewhat random changes to the system and an evaluation of the revised system's fitness for its function. What these methods learn is hidden in the numerical values of the parameters: it is difficult to give a description of what has been learned and it is difficult for the systems to represent their own learning to facilitate further progress. Convergence to a solution can be slow and the method is susceptible to the 'hill-climbing problem', that is, it may converge to a locally optimal solution and miss a globally better one, because it cannot make the large changes needed to reach it. One technique to escape from a local maximum, called 'simulated annealing' because of an analogy with the annealing process with metals, involves taking random steps away from this local maximum. Overall, though, the disappointing results may lead to a conclusion that you need a lot of knowledge to learn a little more.

Much of our learning seems to occur by reasoning about particular positive instances and negative instances of some concept, rather than by some statistical analysis of very large numbers of instances. We try to induce

an explanation of why some instances are positive and some are negative and hence come to an understanding of the concept:

> **This evening, my son and I embarked upon a pleasant excursion to collect examples of the wild flowers with which this part of the forest is so abundantly blessed. We collected a daisy, and fifty-nine things that weren't.**
>
> *Alan Coren (1979), The unnatural history of Selbourne, Punch, June 6.*

If we observe positive and negative examples of, say, a daisy then `p1 or p2 or ... pn` would be a representation of the concept of daisyness consistent with our observations, where `pi` is a description of the i'th positive instance. This, however, only enables us to lookup a new instance: if we have already observed something with the same description as a positive instance then our concept representation would return true, meaning, yes, it is a daisy, otherwise it would not. If we generalised the concept to, for example, 'a daisy is a small yellow flower' then we would be able to categorise previously unseen instances.

But of course this generalization is at a cost for it does not logically follow from the observations and may misclassify new instances:

> **All generalizations are false, including this one.**
>
> *Alexander Chase (1966), Perspectives.*

Alexander Chase is correct: not all generalizations are false. All generalizations *may be* false, because they attempt to extrapolate from the seen to the unseen. But sometimes, glory be!, we are lucky and the generalization turns out to be true. Nonetheless, there persists a feeling that, like all learning processes it seems, generalising is a dangerous activity:

> **General and abstract ideas are the source of the greatest errors of mankind.**
>
> *Jean-Jacques Rousseau (1762), Emile.*

Rousseau's *Emile* presented an idealistic argument that all learning should come from direct experience of the world and not from books or instruction. It is the one of the earliest attacks of the romantics against the classical rationalism that was becoming increasingly dominant. Generalization may, of course, lead to error but it may also provide insightful and useful conceptualisations.

The process of generalization or induction (that is, going from specific instances to a general rule) is of a different nature to that of deduction (that is, going from general rules to a specific conclusion, as discussed earlier). We could attempt to formalise this informal description of the two processes as inverses of one another. For example, if we represent a general statement

such as "If an Australian meets someone he says 'Gday'" in predicate logic by

```
Australian(x) and Meets(x, y) → Says(x, Gday)
```

then we can deduce

```
Australian(Bruce) and Meets(Bruce, Sheila)
   → Says(Bruce, Gday)
```

that is, if a particular Australian, Bruce, meets Sheila he says Gday.

Now, if we had observed this particular event without having the general rule, then we might have *induced* the general statement, by an inverse process of replacing particular names with variables. However, it is clear that, while this captures something of the nature of induction, it is fraught with difficulty. For a start, how do we know which are the right predicates to describe the event? The method assumes we already know the form of the induced rule, because no new predicates can be introduced. How do we know which specific names to replace by variables? For example, why could we not have induced

```
Australian(x) and Meets(x, Sheila) → Says(x, Gday)
```

that is, "If an Australian meets Sheila he says Gday"? Could we generalise over predicates, to form

```
P(x) and Meets(x, Sheila) → Says(x, Gday)
```

that is, "If anybody, of any nationality, meets Sheila he says Gday"? Anyway, the point is that the AI methodology of trying to impose formal rigour – and not just on the process of induction – rapidly descends into technical questions that may miss the broad issues. Of course, some people would argue that it is exactly AI's mission to impose precision on vague discussions.

Inductive concept learning methods are distinguished along two dimensions: the degree to which they are supervised (that is, whether the instances are labelled or not as positive or negative) and to which they are incremental (that is, whether they learn as they encounter instances one by one or wait until they have all the instances). Supervised, incremental learning corresponds to a human learner developing a partial concept that is gradually refined by further examples, perhaps presented by a teacher. Unsupervised learning involves the learner trying to develop a useful conceptualisation of a set of observations without them being labelled in any way, perhaps like a scientist trying to form a theory to explain some experimental results.

Inductive logic programming is an advanced application of the idea of using inverse resolution for induction to the problem of developing a theory to explain observations. The theory is expressed as a logic program. Most

methods begin with an empty theory and attempt to learn a set of logical expressions that conform to the data. 'Theory revision' methods work by modifying an initial theory, or one developed incrementally by the system, in the light of new data. Typically, the methods try to make the smallest change necessary to the set of expressions in the theory, by adding or deleting parts of one expression or maybe creating a new one. Naturally, theory revision is only feasible if the initial theory is a close approximation to the final desired theory.

An intriguing aspect of inductive logic programming is that if we consider inverse resolution:

```
   q or r
(p or q) and (not p or r)
```

we can see that it introduces a new predicate `p`. This `p` can be anything at all (because, whatever it is, it would be eliminated by resolution anyway). The trick – and this is the point of inductive logic programming – is to create a predicate that will elegantly explain the observations at hand. Some will see this as the essence of the scientific theory formation process. At least, we can see this invention, or at least introduction, of a predicate as another indication that computers are not necessarily limited to what has been explicitly provided by us.

Induction is logically unsound but obviously necessary to survive in the real world: if the first time we see a snake it bites someone who then dies, it would be wise to assume that all snakes are dangerous. If, however, we consider that all scientific theorising has an inductive nature, in that one makes observations from which one aims to infer general laws, then we have to concede that scientific theories are not logical conclusions:

> **Now in my view there is no such thing as induction. Thus inference to theories, from singular statements which are "verified by experience" (whatever that may mean) is logically inadmissible. Theories are, therefore, never empirically verifiable ... but ... it must be possible for an empirical scientific system to be refuted by experience.**
>
> *Karl Popper (1934), Logik der Forschung, published in English as The Logic of Scientific Discovery (1959), London: Hutchinson.*

Popper concluded that there was no such thing as induction by himself inducing from the fate of classic scientific theories such as those of Galileo and Newton. Clearly, there is such a thing as induction – it is just not logically sound, as perhaps philosophers of science imagined before Popper enlightened them. According to Popper, science is distinguished from non-science not by its theories conforming to the observational evidence but by

the theories being capable of being falsified by some conceivable observation.

The errors made as a result of an over-generalisation are not necessarily a cause for despair: they may be an opportunity to specialise the partly learned concept:

> **We make progress if, and only if, we are prepared to learn from our mistakes.**
>
> *Karl Popper (1963), Conjectures and Refutations: The Growth of Scientific Knowledge, London: Routledge & Kegan Paul.*

It is a cliché that we learn from our mistakes and indeed most AI learning programs can similarly be said to be 'failure-driven'. This seems sensible: if a program makes a good decision, say, makes a good chess move or correctly identifies a daisy, then it seems perverse to change it but if it makes a mistake, for example, says an object is a daisy when it isn't or vice versa, then the program should change the basis for its decision. One of the better known of the many techniques that have been devised is the 'version space' method for concept learning, which involves the program maintaining two bounds, an upper bound representing a maximally specific set of descriptions and a lower bound representing a minimally general set of descriptions, and using positive and negative instances that the program has categorised wrongly to narrow the space enclosed by the bounds to converge on a concept description.

Nonetheless, this process of continually refining a concept on the basis of individual observations still seems a laborious one. How many daisies do we need to see to grasp the concept of daisyness? Often in teaching it seems possible to explain a concept by means of one or more carefully chosen examples. We could, for example, use analogy whereby the concept that is to be learned (the 'target', for example, electricity) is mapped onto a concept that is better understood (the 'source', for example, water). The learning process involves mapping corresponding features from the source to the target. Clearly, which features to map and how to map them may not be obvious, and the method therefore has an inductive rather than deductive nature:

> **Analogies, it is true, decide nothing, but they can make one feel more at home.**
>
> *Sigmund Freud (1933), New Introductory Lectures on Psychoanalysis, New York: W.W. Norton & Co.*

Freud's intervention at this point enables us to comment that his theory of psychoanalysis is a prime example of what Popper would consider to be non-science, because its experimental data, being confidential, are not open to

scientific scrutiny and its limitless analysis of the unconscious does not provide falsifiable predictions.

Analogy is a non-deductive form of reasoning that can be applied to more than just learning. A broader form of analogical reasoning is 'case-based reasoning', which involves the storing of solutions to problems so that when a new problem is encountered it may be compared to the previous problems. A case sufficiently similar is selected and adapted somehow to suit the new problem, a process approved by the recognition of 'case laws', which are laws established by following judicial decisions in earlier cases.

A deductive form of analysis of a single example, 'explanation-based generalisation' (EBG), has become an established machine learning technique. The term EBG itself is a particularly inept terminologisation, even by AI's standards, since there is no explanation or generalisation, as the terms are usually understood. The 'explanation' is really just a proof, showing how the example is an instance of the concept by reasoning from the system's domain knowledge. The 'generalisation' does not enable the system to answer any more questions than it could before, as you would expect from a generalisation, but the system becomes more efficient because it does not need to re-prove the example. The EBG method requires the specification of four things: a domain theory (a set of rules relating to the concept to be learned), an example (an instance of the concept), the conditions that the required concept definition must satisfy, and a rough definition of the concept. The outcome is a more precise definition of the concept derived by proving how the instance is a positive example of the concept.

The development of EBG also illustrates the interplay, which is pervasive in AI, between psychological and computational issues and between formal and informal methods. Initially, EBG had a vague psychological motivation – a pioneer of the method admitted to an:

> **… intuition that real-world human adult learning is largely explanation driven but no psychological experiments have as yet been performed to test this hypothesis.**
>
> *Gerald Dejong (1986), An approach to learning from observation, in Ryszard Michalski, Jaime Carbonell and Tom Mitchell (eds.), Machine Learning II, Los Altos, Ca.: Kaufmann.*

The initial, rather messy, attempts to computerise this intuition were then cleansed by a unifying formalisation which showed that the core of the process was similar to a technique of logic programming called partial evaluation. However, some researchers considered this formalisation to miss the essence of the process.

A standard illustration of a more informal form of EBG concerns the learning of a concept such as ‘kidnapping’ from a newspaper paragraph by reasoning about the commonsense knowledge implicit in the story (that parents tend to value their children, that captured people are held under duress, and so on) and by removing specific details (such as names, the amount of the ransom, the location, and so on). The power of explanation-based learning derives from the knowledge provided to the system to enable the analysis process and is thus part of a general trend in AI towards knowledge-based systems.

The potential utility of having a computer learn concepts such as kidnapping had previously been indicated by a proposed novel combination with another machine learning technique, that of data mining, which involves trying to extract knowledge from very large databases:

> **“This is the Computer Data Bank. Leave $100,000 in small bills in locker 287 at the Port Authority Bus Terminal or I’ll print your complete dossier and send it to your wife.”**
>
> *Saturday Review (January 11 1969), Computer phoning a customer in a cartoon by Henry Martin.*

Data mining is now one of the more successful applications of machine learning, with the rapid growth of on-line data. Typically, large numbers of records (for example, of medical histories or credit card transactions) are analysed to predict subsequent events (such as the effectiveness of some treatment) or to detect anomalous data (such as fraudulent credit card transactions). For example, NASDAQ uses a surveillance tool to analyse thousands of transactions daily in order to help detect insider trading. As always, there is scope for ethically dubious uses, for example, to pester potential buyers about new products that a database analysis indicates they might be interested in.

As far as practical machine learning is concerned, there has been a recent trend away from the symbolic structural approaches of classical AI towards more mathematical and statistical methods, such as ‘support vector machines’, which are derived from neural information processing, computational learning theory, and pattern recognition. We have noted similar trends in game playing, planning, and natural language processing. Perhaps it is inevitable that, as a field develops and the demand for academic status and useful outcomes grows, there is a consolidation on a formal style and tractable problems, with mathematicians moving in to polish up the initial attempts at rigour. Often, this is at the cost of sanitising the original problem. So, today, most machine learning theory and practice assumes the problem to be that of taking a training set of labelled instances and determining a rule for

labelling new instances. So-called probably approximately correct (PAC) learning is an approach that, given a tolerance for errors, determines a hypothesis that is correct with a certain probability, assuming that the training set has the same probability distribution as the test set. This is a nice formal problem but it doesn't bear much relation to what we normally think of as learning.

Those with a psychological orientation can, of course, continue to develop models of human learning, as EBG was initially considered to be. For example, another psychological approach is based on the idea of 'chunking', that is, that two commonly co-occurring items tend to coalesce. For example, if we often use a rule that 'a causes b' followed by one that 'b causes c' then we might chunk these two rules to form 'a causes c'. This kind of process helps to explain how we become more efficient at some task that at first requires the deliberate performance of several different actions, such as driving a car:

> **Civilization advances by extending the number of important operations which we can perform without thinking about them.**
>
> *Alfred North Whitehead (1911), quoted in W.H. Auden (1970), A Certain World: A Commonplace Book, New York: Viking Press.*

Chunking is the basic mechanism of Soar, a descendant of GPS, which aims to explain the nature of general intelligence:

> **One central hypothesis is that chunking, a simple experience-based learning mechanism, can form the basis for a general learning mechanism. Soar uses a production system to encode its knowledge base. Chunking creates new productions (chunks), based on [its experiences], and adds them to the production system.**
>
> *Milind Tambe and Allen Newell (1987), Some chunks are expensive, Proceedings of the Fifth International Conference on Machine Learning, Ann Arbor, MI.*

Soar began life as an uninspired acronym (for State, Operator and Result) but matured, left home and lost all but one of its capitals. Soar is intended to be a general cognitive architecture and is notable for making learning the core process and for emphasising that learning is goal-oriented and does not occur in a vacuum, as some AI learning mechanisms suggest.

The chunking operation works by summarising the information uncovered during a problem solving process and is therefore similar to explanation-based learning. As with GPS, tasks in Soar involve the achievement of goals within problem spaces consisting of sets of states and operators. The basic process is repeatedly to propose, select and apply operators to a state. If an operator cannot be determined, a sub-goal with its own problem space is generated. This sub-goal disappears if it becomes

irrelevant or if it is achieved, and in the latter case the process that led to success is summarised as a new chunk, which will be invoked in similar future situations, thereby more directly producing the required result.

The knowledge representation in Soar and many other learning mechanisms is the production system. The initial use of production systems was to model human problem solving and then to provide efficient expert system performance. Some of the characteristics which suit production systems to these roles also provide the capability for self-modification and hence of learning. The rules are relatively independent of one another and therefore it is possible to insert or delete rules, or change existing ones, without too much concern for interactions with other rules. We can imagine this as being useful to model the incremental, developmental nature of human learning. In addition, the rules themselves have a standardised format, involving sets of conditions and actions, such that it is relatively easy to change the rules, by adding, deleting or modifying individual conditions or actions. Therefore, if a rule has an action that involves such a change to any rule in the system, then we have a system capable of self-modification or learning.

All these methods of learning rely on there already existing something from which to learn – descriptions of observations, commonsense knowledge, rules to chunk, and so on. Often these seem to assume the very thing that is to be learned – how do you know how to describe a 'daisy' unless you have some idea what the concept is? How do you know which of the thousands of potential features are actually relevant? If we know what a daisy is we can easily underestimate this difficulty – obviously, size and colour are relevant. But what about smell, where it grows, when it grows, the shape of the leaves, and so on? Of course, most concepts (such as 'foreigner', 'terrorist', etc.) are not defined in terms of visible features and it may be unwise to assume that they are. Also, an acceptable definition of a concept depends upon the use to which it will be put, which is hard for a system to judge. For example, a child and a botanist will have different but adequate understandings of daisyness.

37. Connectionism: "to escape the brittleness"

In the 1980s, because of continuing difficulties with knowledge-based learning methods and because of new theoretical results, the perceptron and related devices made a remarkable comeback to become the dominant topic at cognitive science conferences. More subtle ways of connecting the devices were developed to overcome some of the theoretical limitations of simple

perceptrons. In part, the continuing fascination with perceptron-like devices comes from a feeling that any effective computational learning mechanism will need to work much like the way we learn and there is, of course, a superficial analogy between the highly interconnected, simple devices of a perceptron and the neurons of 'neural networks' that have been studied in the human brain.

It is natural to hope to design computer systems which learn by using methods by which the most advanced learners known (that is, humans) have learned, and in the 1960s there was hope that the idea of evolution could be adapted to enable programs to improve themselves:

> **The promise of artificial evolution is that many things are known or suspected about the mechanisms of natural evolution, and that those mechanisms can be used directly or indirectly to solve problems in their artificial counterparts. For artificial intelligence research, simulation of evolution is incomparably more promising than simulation of neural nets, since we know practically nothing about natural neural nets that would be at all useful in solving difficult problems.**
>
> *Ray Solomonoff (1966), Some recent work in artificial intelligence, Proc. of the IEEE, 54, 12, 1689.*

> **In nothing is there more evolution than the American mind.**
>
> *Walt Whitman (1881), Notes Left Over, Foundation Stages – then Others.*

Unfortunately, evolution is an exceedingly slow process, although, of course, the hope is that a computer simulation will speed it up to enable useful outcomes within a reasonable time.

It is helpful to have an idea of how slow. Every organism has a set of genes, called its genotype, and each gene can exist in several forms, called alleles. The organism is an amalgamation of characteristics, each of which is determined by one or more genes. Evolution is a search for high points in a multi-dimensional field of genotypes corresponding to optimal degrees of adaptation to the environment. If the genotype has 10,000 genes, each with two alleles, then there are $2^{10,000}$ forms to search and, even worse, because of epistasis – that is, the fact that the effect of one allele can depend on what other alleles are present – very similar genotypes can have very different degrees of fitness to their environment. (There are, in fact, 13,601 genes in the Drosophila fruit fly, one of the first animals to have its genomes decoded. The Human Genome project ended in April 2003 with an estimate of 30,000 to 40,000 for the number of human genes, which is much fewer than had been anticipated.)

There is some irony in the fact that advocates of computational evolution were reacting to the perceived deficiencies of standard search-

based AI because evolution is, of course, a blind search with no guarantee to reach optimal solutions. Advocates of behaviour-based AI, arguing that the route to AI should be bottom up, via robotic insects, reptiles, and so on, point out that evolution took billions of years to produce insects and only a fraction of that to move on to humans:

> **This suggests that problem solving behavior, language, expert knowledge and application, and reason, are all pretty simple once the essence of being and reacting are available. That essence is the ability to move around in a dynamic environment, sensing the surroundings to a degree sufficient to achieve the necessary maintenance of life and reproduction. This part of intelligence is where evolution has concentrated its time – it is much harder.**
>
> *Rodney Brooks (1991), Intelligence without representation, Artificial Intelligence, 47, 139-159.*

However, this advocacy is somewhat undermined by the research programme starting with robotic insects rather than primeval sludge.

So-called 'genetic algorithms' take a representation of an 'individual', usually as a sequence of bits, make changes to generate offspring individuals, and then apply a 'fitness function' to select the most successful individuals for the next generation. A change is generally a mutation (that is, a random alteration to a single bit) or a cross-over (that is, the result of merging the first part of one individual with the second part of another). Since the representations are inscrutable, the mutations are randomised, and the system makes no attempt to find relationships between individuals and success, it is hard for human observers to keep track of what is happening during such a computational evolutionary process:

> **[The evolutionary approach] has had no substantial success so far, perhaps due to inadequate models of the world and of the evolutionary process, but it might succeed. It seems dangerous since a program that was intelligent in a way its designer didn't understand might get out of control.**
>
> *John McCarthy and Patrick Hayes (1969), Some philosophical problems from the standpoint of artificial intelligence, in Bernard Meltzer and Donald Michie (eds.), Machine Intelligence 4, New York: American Elsevier.*

If a little learning is a dangerous thing then a lot of evolution must be. Is it possible that the designer of natural evolution also does not understand the intelligence He has created?

Genetic algorithms are particularly opaque but in this respect are not fundamentally different from any algorithm of the complexity necessary for intelligence. McCarthy and Hayes, being logicians, underestimate the incomprehensibility of other notations, such as predicate logic. If all systems

whose intelligence its designers could not understand were prohibited as dangerous then AI research would be stillborn, as indeed some argue that it should be.

The resurgence of perceptron-like learning mechanisms, under the names of connectionism, neural networks, parallel distributed processing and adaptive networks, was not entirely driven by theoretical or philosophical breakthroughs. Academics prefer to believe that changes in research directions are the result of their own profound analysis and reasoning – but sometimes they are driven by technology:

> **Recent technological advances in VLSI and computer aided design mean that it is now much easier to build massively parallel machines. This has contributed to a new wave of interest in models of computation that are inspired by neural nets rather than the formal manipulation of symbolic expressions. To understand human abilities like perceptual interpretation, content-addressable memory, commonsense reasoning, and learning it may be necessary to understand how computation is organized in systems like the brain which consist of massive numbers of richly interconnected but rather slow processing elements.**
>
> *Geoffrey Hinton (1989), Connectionist learning procedures, Artificial Intelligence, 40, 185-234.*

As situated cognition would suggest, researchers find justifications for the research that is appropriate for the situation they find themselves in. Indeed, situated cognition itself only became fashionable when technology enabled distributed processing and the kind of community-based activities that situated cognition considers more important than the individual cognition simulated by personal computers.

There are many forms of connectionist model but they generally describe a parallel architecture, inspired by the neural structure of the brain, in which simple processors are interconnected by weighted links. The long-term knowledge held by connectionist networks exists in the strengths of the connections between nodes. The networks learn by the gradual modification of the weights, enabling the system to perform a given task. Although there are similarities with the coefficient modification methods of, for example, Samuel's program, connectionist systems generally improve on symbol-processing learning systems in being more robust in handling unreliable data and degrading more gracefully. On the other hand, of course, they lack explicit knowledge representations, making it difficult for users to understand what has been learned and to modify it.

In the simplest case, the network (equivalent to a simple perceptron) consists only of inputs, outputs and the weighted links between them. More generally, one or more layers of 'hidden units' connect the inputs and outputs:

… Your hip bone connected to your back bone,
Your back bone connected to your shoulder bone,
Your shoulder bone connected to your neck bone,
Your neck bone connected to your head bone,
I hear the word of the Lord!

Anon, Dem Dry Bones.

The optimal number of hidden units is difficult to determine, it being a compromise between the need to minimise redundant links and yet have sufficient to learn what is required. In 'feed-forward' networks, the links flow from the inputs to the outputs.

The most popular learning mechanism in such networks is 'back-propagation', which adjusts the weights on links to output units in a way similar to coefficient modification and then propagates those changes to the links on hidden units connected to the output units and so on to the input units. The back-propagation algorithm was developed by David Rumelhart and colleagues in 1985 and caused a frenzy of research in neural networks similar to that provoked by the development of resolution in the theorem proving field two decades earlier. Actually, the history of the algorithm is rather confused by it being ignored in the aftermath of the Minsky and Papert critique but the idea of it seems to have been mentioned by Frank Rosenblatt in 1962, made precise by Arthur Bryson and Yu-Chi Ho in 1969 and then re-discovered by Paul Werbos in 1974. The intense research led to the development of many sophisticated mathematical techniques to propagate the changes 'fairly' to the links. Indeed, connectionism has become a battlefield for mathematicians, physicists, statisticians and biologists, with the initial psychological motivation somewhat neglected, and perhaps just as well:

But is this what the brain does? Alas, the back-[propagation] nets are unrealistic in almost every respect, as indeed some of their inventors have admitted ... Most of these neural net 'models' are therefore not really models at all, because they do not correspond sufficiently closely to the real thing.

Francis Crick (1989), The recent excitement about neural networks, Nature, 337, 129-132.

Francis Crick (1916-2004) was co-winner of a Nobel Prize in 1962 for discovering the structure of the DNA molecule and later carried out research at the Salk Institute, California into how brain neurons are activated by what we see.

The fact that the knowledge of a concept within a connectionist network is distributed over many units makes it difficult to say what each unit represents, if it can be said to represent anything. Such networks are therefore very different from Bayesian networks: in the latter, nodes of the network are intended to have symbolic meaning; in the former, the 'meaning' lies, if anywhere, in the weights, with nodes forming abstract connections. Unlike standard expert systems, it is not easy to provide an intelligible explanation of how a conclusion has been derived from a neural network. Displaying a large set of weights hardly qualifies as an explanation. However, it is possible to derive theoretical results, based on the training sequence, concerning the probable correctness of conclusions so that users may have some degree of trust in, if not comprehension of, the system. Connectionist researchers recognise the poor expressive power of networks compared to, say, statements in predicate logic, and some of them aim to add structure to the networks to improve their representational power. Others, however, consider the two methodologies complementary, each contributing benefits to help overcome weaknesses of the other:

> **Connectionist models may well offer an opportunity to escape the brittleness of symbolic AI systems, a chance to develop more human-like intelligent systems – but only if we can find ways of naturally instantiating the sources of power of symbolic computation within fully connectionist systems.**
>
> *Paul Smolensky (1990), Tensor product variable binding and the representation of symbolic structures in connectionist systems, Artificial Intelligence, 46, 159-216.*

In other words, probably the main virtue of neural networks lies in their tolerance of errors in the input data. Neural networks do the best they can to fit the data provided. If the data is in error, performance degrades gracefully, unlike for conventional symbolic processing, which, as we have seen, tends to have a true-or-false view of the world.

In retrospect, the great conflict between connectionism and symbol-processing (considered to derive from the Turing machine view of computation) is amusing if one re-reads Turing's *Intelligent Machinery* paper of 1948. This is, in fact, mainly concerned with machine learning and connectionism, although, of course, he did not use the term. He wrote about 'unorganised machines' consisting of networks of neuron-like Boolean elements connected together in a largely random manner. In the last years of his life he worked on modelling biological growth (which today would be called artificial life research). So Turing himself saw no conflict. The original 1948 paper was not released for twenty years because his superior had considered it unsuitable for publication. It evolved into the paper

published in *Mind* in 1950 – one of the most widely discussed papers ever written.

38. Creativity: "the envy of other people"

Somehow, the methods for computerised learning do not seem to get to the crux of the matter – learning, we might feel, is about coming up with something original, something that is not implicit in what is already available. At least, we might argue that what is learned should be new, in some sense, for the individual learner, if not for all humankind.

The fact that computers cannot be original was stated at the start:

> **The Analytical Engine has no pretensions whatever to originate anything. It can do whatever we know how to order it to perform.**
>
> *Ada Lovelace (1843), Translation and notes of a Sketch of the Analytical Engine invented by Charles Babbage, by Louis Menebrae, in Richard Taylor (ed.), Scientific Memoirs, Selections from the Transactions of Foreign Academies and Learned Societies and from Foreign Journals.*

While it is literally true that computers can only do whatever we know how to order it to perform, it does not follow that we cannot order it to originate something.

Where do new ideas come from? Is it an entirely mystical process that we cannot hope to pin down sufficiently to enable us to write a program to carry out?:

> **All the good ideas I ever had came to me while I was milking a cow. So I went back to Iowa.**
>
> *Grant Wood (1932).*

That's promising. We now have robotic milking machines that identify the cow, wash her, locate the teats, milk until the flow drops, record the yields, and could surely ruminate all the while. Farmers, like the old cow, are over the moon about them. And thankfully not all cows are in Iowa.

Or is the creation of new theories dependent on the prior disinterested accumulation of numerous facts that can then be subjected to some kind of scientific analysis?:

> **After my return to England it appeared to me that by ... collecting all facts which bore in any way on the variation of animals and plants under domestication and nature, some light might perhaps be thrown on the whole subject ... I worked on true Baconian principles, and without any theory collected facts on a wholesale scale.**
>
> *Charles Darwin (1876), The Autobiography of Charles Darwin, edited by Nora Barlow.*

Well, computers should be good at the mindless collection of data and indeed there are databases of previously inconceivable size on all sorts of topics. But how does a computer (or a person) know which facts to collect without some basis or "any theory", as Darwin said, for selecting among them? Look around now and decide which of the infinity of 'facts' visible you would write in a notebook.

Or is the generation of new ideas based on the systematic combination of old ideas?:

> **The fact is that in admiring the productivity of genius our admiration has been misplaced. Nothing is easier than the generation of new ideas: with some suitable interpretation, a kaleidoscope, the entrails of a sheep, or a noisy vacuum tube will generate them in profusion. What is remarkable in the genius is the discrimination with which the possibilities are winnowed. A possible method, then, is to use some random source for the generation of all the possibilities and to pass its output through some device that will select the answer ... We can now proceed to build the system whose selectivity, and therefore whose intelligence, exceeds that of its designer.**
>
> *W. Ross Ashby (1956), Design for an Intelligence Amplifier, in Claude Shannon and John McCarthy (eds.), Automata Studies, Princeton, N.J.: Princeton University Press.*

Computers should have a real advantage here. Lacking the common sense to know better, they will happily combine old ideas in all sorts of novel formations. However, there are just too many possibilities to generate them all at random and then somehow sieve out the successful ones.

Still, there is reason for optimism as we aim for computer creativity. Or perhaps we shouldn't: some people insist that they prefer their machines to be reliable and consistent. The last thing they want is machines that are unpredictable and are liable to interfere with flashes of creative inspiration. However, creativity seems to be a component of human intelligence and an essential attribute of artificial intelligence, whatever it is precisely:

> **Creativity is just the envy of other people.**
>
> *Marvin Minsky (1986), The Society of the Mind, New York: Simon and Schuster.*

As there are many other characteristics, such as wealth, health, charisma, honesty, and so on, that we might envy, this has the appearance of the kind of glib but vacuous definition of a difficult concept that AIers are fond of, but it makes a deep point. Creativity is something that we attribute to other people but they themselves would not necessarily consider themselves creative. To them, the creative act is just the inevitable and unremarkable outcome of their experience and expertise. Somehow, they have the ability to focus directly on those possibilities that are most likely to be successful,

rather than consider them all in sequence, as Darwin and Ross Ashby proposed:

> **There exist too many combinations to consider all combinations of existing entities; the creative mind must only propose those of potential interest. ... The true work of the inventor consists in choosing among these combinations so as to eliminate the useless ones or rather to avoid the trouble of making them, and the rules which must guide this choice are extremely fine and delicate. It is almost impossible to state them precisely; they are felt rather than formulated.**
>
> *Henri Poincaré (1929), The Foundations of Science: Science and Hypothesis, the Value of Science, Science and Method, New York: The Science Press.*

Clearly, if the 'rules' for invention, fine and delicate though they may be, are really "felt rather than formulated" then there is little prospect of being able to write a program to execute them.

Perhaps the impressiveness of the results of invention, like the magicians' rabbits, masks the simplicity of the processes that led to them:

> **The creative act is not an act of creation in the sense of the Old Testament. It does not create something out of nothing; it uncovers, selects, re-shuffles, combines, synthesizes already existing facts, faculties, skills. The more familiar the parts, the more striking the new whole.**
>
> *Arthur Koestler (1967), The Act of Creation, London: Picador.*

This view seems to be no more than a paraphrase of that put forward by the Scottish philosopher David Hume (1711-76):

> **... all this creative power amounts to no more than the faculty of compounding, transposing, augmenting and diminishing the materials afforded us by the senses and experience.**
>
> *David Hume (1748), An Enquiry Concerning Human Understanding.*

If creativity only involves shuffling and re-combining known facts then computers have a chance, so, in that spirit, let us see what such processes can accomplish.

39. Discovery: "an irrational element"

AM, a notorious AI program devised by Douglas Lenat, was supposed to discover or create a theory to account for observed data (which happened to be in the field of mathematics). It was provided with about 250 allegedly general heuristics for determining what was interesting to investigate and about 100 elementary mathematical concepts, said to be those possessed by the average four-year-old child. The general heuristics included, for example:

```
If f(x,y) is an interesting function then consider f(x,x).
If g(x,s) is an interesting function mapping x onto a set s
   then consider those x which are mapped onto large s's.
```

It is a challenge to imagine what the other 248 general heuristics might be. Being general, they should be applicable not just to mathematical concepts but also to all other concepts, including everyday ones. So, for example, once the concept of murder is considered interesting, the following rules might be applied:

```
If murder(x,y) is an interesting function then consider
   murder(x,x).
If murder(x,s) is an interesting function mapping x onto a
   set s then consider those x which are mapped onto large
   s's.
```

This might be considered to amount to the 'discovery' of the concepts of suicide and multiple murderer.

After an hour or so, AM discovered mathematical concepts such as prime number, Diophantine equations, Goldbach's conjecture and maximally composite numbers. If you do not know what these concepts are, you will be suitably impressed. But of course you cannot judge to what extent those concepts were seeded by the general heuristics and elementary concepts already provided. To do so you would have to look closely at all their definitions within AM to see whether its pretensions to originate anything are reasonable:

> **Since the heuristics did lead to the discoveries, they must in some sense be an encoding for them, but they are not a conscious or (even in hindsight) obvious encoding. Skepticism of a program's generality is necessary and healthy. Is AM's knowledge base 'just right' – that is, finely tuned to elicit this one chain of behaviors? The answer is 'No!'.**
>
> *Douglas Lenat (1983), The role of heuristics in learning by discovery, in Ryszard Michalski, Jaime Carbonell and Tom Mitchell (eds.), Machine Learning, Palo Alto: Tioga.*

Some people thought that Lenat protested too much. Sufficient scepticism remained that accusations of dubious practices were made:

> **Many rules contain 'special purpose hacks' which Lenat does not describe. Unfortunately, some of AM's most interesting results depend crucially on such rules. It can easily be conjectured that these hacks were written for the special purpose of making just these rediscoveries.**
>
> *Jon Rowe and Derek Partridge (1993), Creativity: a survey of AI approaches, Artificial Intelligence Review, 7, 43-70.*

The main reason for the program's notoriety lies not in its achievements or its failures but in the research methodology that it represents.

AM is a very complicated program and its description was accompanied by grandiose statements about its capabilities. The scientific community could not verify those statements in any objective fashion. Even simple experiments were not or could not be carried out. For example, using a different set of initial concepts could have tested claims that the heuristics were general. To test that no particular heuristic or concept was crucial to the program's performance (in other words, that the results were not a fairly direct consequence of a particular piece of given knowledge and hence hardly original), the program could have been re-run with some of the heuristics or concepts deleted. Some of these kinds of study were made but only in a half-hearted, unconvincing way.

Most sciences progress through independent sceptical scientists testing a proposed theory and AI should be no exception. A difficulty for AI is that the theory is embedded in a program that is the outcome of many years' work and that its owner is naturally reluctant to make freely available.

In one sense, the program *is* very impressive: in one hour it apparently discovered concepts that it took the whole human race millennia to discover. It certainly impressed its author by discovering mathematics that he didn't know about:

> **AM was not able to discover any 'new-to-mankind' mathematics purely on its own, but has discovered several interesting notions hitherto unknown to the author.**
>
> *Douglas Lenat (1983), The role of heuristics in learning by discovery, in Ryszard Michalski, Jaime Carbonell and Tom Mitchell (eds.), Machine Learning, Palo Alto: Tioga.*

On the other hand, it missed some important concepts, such as real numbers, fractions, and infinity.

An obvious question is: why was AM stopped after only one hour? The answer appears to be that it ground to a halt – that is, its general heuristics were too general to be useful with the increasingly advanced concepts created. Clearly, new heuristics needed to be created as well, but that is easier said than done - although Lenat said that he had done it in his following system, called EURISKO, by treating the heuristics just as rules, like any other, and applying the heuristics to themselves, that is, to discover new heuristics.

On reflection, it was conceded that the apparent success of AM was mainly due to the fact that syntactic mutations of Lisp programs tended to be meaningful because of the mathematical basis of Lisp and it happened to be

mathematics that AM was seeking to discover. The implications for a program to discover heuristics or anything else are clear – the mutations would need to be applied to a notation appropriate for that topic, if one can be devised.

With that humility for which AIers are renowned, Lenat speculated from the apparent success of his programs at discovery learning and the apparent failure of others' programs intended to simulate evolution that perhaps evolution did not work the way everyone had thought, from random mutations. Maybe evolution worked more like EURISKO:

> **Nature might already have become as good at programming in the last billion years as we have in the last forty. DNA might have *already* evolved from random generate and test into an expert program (expert at mutating itself in plausible coordinated ways, expert at designing improved progeny) ... This is how EURISKO uses a set of heuristics to improve and extend itself.**
>
> *Douglas Lenat (1983), The role of heuristics in learning by discovery, in Ryszard Michalski, Jaime Carbonell and Tom Mitchell (eds.), Machine Learning, Palo Alto: Tioga.*

It is always a problem for AIers, after almost solving the greatest problems of mankind, to know what to do for an encore. From the threshold created by AM and EURISKO, Lenat turned aside to the ten-year, $50 million CYC project, mentioned earlier.

This project aimed to develop an encyclopedic knowledge base, by systematising the commonsense knowledge needed to read about a hundred Encyclopedia Britannica entries, after which the system would, it was said, be able to read and systematise the rest for itself. CYC was the most ambitious knowledge representation and knowledge acquisition project ever undertaken. Its ambition is reminiscent of H.G. Wells's concept of a 'World Brain', a universally accessible repository of all human knowledge:

> **The time is ripe for a very extensive revision and modernisation of the intellectual organization of the world … the whole human memory can be, and probably in a short time will be, made accessible to every individual … [the World Brain will be] an efficient index to *all* human knowledge, ideas and achievements … a complete planetary memory for all mankind.**
>
> *H.G. Wells (1937). World Brain: The Idea of a Permanent World Encyclopaedia, Encyclopédie Française.*

Given H.G. Wells' reputation as the father of science fiction and his visions of time travel and space exploration, it ought to be stressed that the World Brain was a serious scientific proposal not a fantasy. Maybe the World Wide

Web has delivered it, if it is considered to be "an efficient index to all human knowledge".

CYC sought to end the brittleness of expert systems by creating a huge knowledge base of 'consensus reality' or common sense. It was estimated that this would require at least ten million appropriately organized items of information, including rules and facts that describe concepts as abstract as causality and mass. A small army of knowledge engineers would enter some of these items and techniques of discovery learning would add many further items. Naturally, this would involve the consideration of abstruse theoretical and philosophical issues relating to the nature of such knowledge, or at least the avoidance of them:

> **We occasionally have had to delve into the quagmires of logic, but we have studiously avoided the quagmires of philosophy. How did we do that, when we had to represent time, substances, perception, contexts, belief, and so on? The philosophical quagmire around such topics has had three thousand years to form. We avoided it by being pragmatic.**
>
> *R.V. Guha and Douglas Lenat (1993), Re: CycLing paper reviews, Artificial Intelligence, 61, 149-174.*

However, pragmatism is itself an honourable philosophical theory of meaning, introduced by the American philosopher Charles Peirce in 1878. He argued that our conception of an object is precisely the conceivable practical bearings that we consider the object to have. This view was subsequently adopted and distorted by others – to Peirce's annoyance, for he tried to re-name the original theory as 'pragmaticism', a word "ugly enough to be safe from kidnappers" but also, it proved, too ugly for anyone to use at all. Pragmatism now means little more than an emphasis on results alone as a measure of success. Therefore, it might seem best to move immediately to the success or otherwise of CYC.

It is interesting, however, to see how CYC evolved during its ten years, for it is one of the best examples of the struggle between the AI 'neats' (those who believe in the need for the rigour of logic-like notations to formalise intelligent processes) and the 'scruffies' (who don't). During the course of the project, the CYC engineers, perhaps inevitably, increasingly came to use notations suspiciously close to those of predicate logic and to adopt techniques similar to those developed by, for example, commonsense reasoning researchers. Perhaps those quagmires of logic and philosophy cannot be so easily circumnavigated.

CYC funding ended in 1995 or so: did it succeed? In keeping with the pragmatic approach, its designers considered that CYC would be regarded as

a success if the product was used by researchers and engineers in developing new expert systems and, indeed, by all of us:

> **[By 1999] no one would even think of buying a computer that doesn't have CYC running on it.**
>
> *Douglas Lenat and R.V. Guha (1990), Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project, Reading, Mass.: Addison Wesley.*

On that basis, very few computers would be bought today. Some parts of the CYC system were publicly released and others were sold through a company set up for that purpose. Academic writing on the project was roughly proportional to the amount of funding remaining. No final report seems to have been published in a journal, as would be expected for large research projects, and no summary is available on the Web.

To return to the discovery processes of AM, this project indicated that a heuristic search of combinatorial spaces could lead to interesting conjectures but AM was never used for any serious mathematical purpose. The closest practically oriented systems today are those that aim to carry out 'knowledge discovery' (similar to data mining mentioned above) from the large databases now on-line in order to bring to light previously unrecognised knowledge hidden in those databases.

For example, one system, called Arrowsmith, developed by Neil Smalheiser and Don Swanson at the University of Chicago, analyses the Medline database to make conjectures about treatments for medical diseases. If, for example, it finds data suggesting that magnesium is a treatment for spreading depression and that spreading depression is associated with migraine attacks, then it might conjecture that magnesium could be a treatment for migraine. To help ensure that such conjectures are novel, it checks the Medline database to see that nobody has previously reported such a treatment. In this case, subsequent laboratory tests confirmed a link between magnesium deficiency and migraine attacks. In general, the process involves searching two separate literatures, looking for associations between A and B in one and between B and C in another. The hope is that such systems will generate possibilities that scientists have overlooked. It is a feature of the systems that they engage in some kind of human-computer collaboration to focus and refine the conjectures.

Data in the biological sciences, being voluminous, recently generated and mainly symbolic, seems particularly amenable to such approaches. The HAMB system, developed by Gary Livingston, Bruce Buchanan (of DENDRAL fame) and John Rosenberg, carries out an AM-like search of an empirical, rather than mathematical, domain, such as the Biological Macromolecule Crystallization Database of 2255 cases. Its task is to find

conditions for growing good crystals. At the core of HAMB is a rule-induction program that looks for general rules of the form

```
if [conditions] then [class]
```

that fit items in the database. Like AM, it maintains a measure of interestingness for its on-going tasks. Its output shows that:

> **HAMB has been demonstrated to find interesting and novel relationships in published data about crystal-growing experiments.**
>
> *Bruce Buchanan and Gary Livingston (2004), Toward automated discovery in the biological sciences, AI Magazine, 25, 1, 69-84.*

In particular, in one run, HAMB generated 575 rules, 92 of which were considered 'apparently novel and significant' and hence perhaps worthy to be considered discoveries.

A number of systems have been implemented which attempt to replicate the discovery of scientific laws from experimental data. These systems are unconvincing as a demonstration that key aspects of the discovery process have been captured because there is always a suspicion that hindsight has been of assistance. It is of more interest to ask whether such systems have discovered any *new* laws. It seems that the PAULI system, developed by Raúl Valdés-Pérez at Carnegie Mellon University, has enabled the discovery of a new theorem in particle physics, the details of which are less appreciated by us than by particle physicists.

For some people, the AM program and others which aim to 'create' mathematics, physics, music, art, and so on indicate that the creative process is merely the amalgamation of various techniques, such as pattern matching, search, and representation transformations, already well known in AI. All we need to do is tune them to achieve creative excellence:

> **These small acts of creativity, though they differ in scope, are not different in kind from the brilliant leaps of an Einstein. Creativity is commonplace in cognition, not an esoteric gift bequeathed only to a few.**
>
> *Roger Schank and Tom Cleary (1995), Making machines creative, in Steve Smith, Thomas Ward and Ronald Finke (eds.), The Creative Cognitive Approach, Cambridge, Mass.: MIT Press.*

This is a form of argument which may be adapted to many areas of AI: We don't know how some process occurs so we may safely assert that it works basically like some specified simple mechanism, which just needs some unidentified enhancement to achieve desired expert levels of performance.

Most people remain unconvinced. A feeling persists that creativity or discovery is an irrational process, which cannot be simulated by computational rules:

Infallible rules of discovery leading to the solution of all possible mathematical problems would be more desirable than the philosophers' stone, vainly sought by alchemists.

George Polya (1945), How to Solve It, Princeton, N.J.: Princeton University Press.

My view of the matter, for what is worth, is that there is no such thing as a logical method of having new ideas ... My view can be expressed by saying that every discovery contains "an irrational element", or "a creative intuition".

Karl Popper (1934), Logik der Forschung, published in English as The Logic of Scientific Discovery (1959), Hutchinson: London.

If we continue to believe that creative thinking involves the bringing into being of something out of nothing, then computers are necessarily incapable of it, because no computer will ever do anything if we provide it with nothing, that is, if we give it no program of any kind. As soon as we provide a program, however simple, it will always be arguable that whatever is produced is ultimately inherent in that simple program. Of course, the same may be true of people but we prefer to believe otherwise:

The mathematician does not work like a machine; we cannot overemphasize the fundamental role played in his research by a special intuition (frequently wrong), which is not common-sense, but rather a divination of the regular behaviour he expects of mathematical beings.

Nicolas Bourbaki (1950), The architecture of mathematics, American Mathematics Monthly, 57, 221-232.

Taking collaborative working to its natural conclusion, Nicolas Bourbaki is not an individual but a group of mathematicians aiming to revitalise French mathematics. He/they will surely succeed if he/they can provide mathematicians with the power of divination, which the dictionary defines as "the art of discovering future events or unknown things, as though by supernatural powers", to go along with the hotline to truth that we have already met.

Perhaps we should eschew mathematical creativity and stick to more down-to-earth creative activities, such as gardening:

You cannot endow even the best machine with initiative; the jolliest steam-roller will not plant flowers.

Walter Lippmann (1914), A Preface to Politics, "Routineer and Inventor".

Nobody has ever described a computer as jolly, and more's the pity perhaps if creativity, at least of an artistic nature, requires some emotional or passionate engagement.

40. Emotion: "easier than thought"

If creativity, initiative and intuition continue to be attributes that we are reluctant to apply to computers then for the person in the street it appears even more obvious that the human propensity for 'emotional behaviour' cannot sensibly be ascribed to computers. Computers are devices for slavishly carrying out instructions that we give them: they have no affective attitude towards those instructions, whereas, some argue, it is a distinguishing characteristic of human beings that emotions sometimes take precedence over objective rational thought:

> **Man is a creature distinguished not only by the intelligence but by the affections as well ... so ... in confronting the computer, we must examine with care whether the rule is not equally absolute: the society that does not value the educated heart – or wherever the seat of the affections is – will also die.**
>
> *Elting Morrison (1966), Men, Machines and Modern Times, Cambridge, Mass.: MIT Press.*
>
> **Reason guides but a small part of man, and that the least interesting. The rest obeys feeling, true or false, and passion, good or bad.**
>
> *Joseph Roux (1886), Meditations of a Parish Priest.*

The philosophical doctrine of emotivism holds that value judgements express emotions rather than follow from facts, or, as David Hume argued, that there is no logical argument from 'is' to 'ought'. If your value judgement is that value judgements are more important than factual ones then it would follow logically that for you emotions precede facts.

On the other hand, it might be argued that an intelligent being should not allow emotions to overrule conclusions that follow from some rational analysis:

> **The sign of an intelligent people is their ability to control emotions by the application of reason.**
>
> *Marya Mannes (1958), More in Anger, 3.1.*

When Kasparov lost to Deep Blue his reaction was not to concede intellectual superiority – he attributed his failure to an inability to deal with the stress of the situation, an incapacity unknown to Deep Blue:

> **My whole preparation was a failure because Deep Blue played very differently from what I expected. My preparation was based on some wrong assumptions about its strategy; and when after game 2 it proved to be a disaster, I over-worked myself. I actually spent more energy on the games in this match than for any before in my life ... When game 6 finally**

came, I had lost my fighting spirit. I simply didn't have enough energy left to put up a fight. At the end of game 5 I felt completely emptied, because I couldn't stand facing something I didn't understand.

Garry Kasparov (1997), message to Club Kasparov.

Kasparov went on to insist that "this thing is beatable" and that he would win a re-match. But he did not get the chance until January 2003 when in an inconclusive six-game match with Deep Blue's successor, Junior, the two players won one game apiece.

If emotions are our Achilles heel then possibly they stand in the way of our further evolution, to be superseded perhaps by beings without emotion and with superior reason:

The usual course taken by an evolving line has been one of degeneration. It seems to me altogether probable that man will take this course unless he takes conscious control of his evolution within the next few thousand years. It may very well be that mind, at our level, is not adequate for such a task probably on account of its emotional rather than intellectual deficiencies. If that is the case we are perhaps the rather sorry climax of evolution.

J.B.S. Haldane (1932), The Causes of Evolution, New York: Harper and Row.

It is natural to speculate on the causes, if any, of the evolution of emotion. One view is that it provides the longer-term foundation for our shorter-term goals:

The power that possessed him seemed to have nothing to do with reason: all that reason did was to point out the methods of obtaining what his whole soul was striving for …He acted as though he were a machine driven by the two forces of his environment and his personality; his reason was someone looking on, observing the facts but powerless to interfere.

W. Somerset Maugham (1915), Of Human Bondage, London: Heinemann.

If so, then computers might need to develop emotions in order fully to engage in the autonomous, goal-directed behaviour that we assume is characteristic of intelligent beings. This would not imply that computers be emotional in the sense of being excessively affected by its feelings but that its functioning would need to be based on an on-going affective undercurrent.

It seems to be the common opinion that reason, the basis for current attempts to design artificial intelligence, is necessarily in conflict with emotion. Intelligence has to be cold and objective, based on a neutral analysis of the 'facts'; emotion just gets in the way of the desired analysis, or may be the result of being unable to carry out the desired analysis:

The intellect is always fooled by the heart.

Duc de la Rochefoucauld (1664), Les Maximes, 102.

The degree of one's emotion varies inversely with one's knowledge of the facts – the less you know, the hotter you get.

Bertrand Russell.

Man is a rational animal who always loses his temper when he is called upon to act in accordance with the dictates of reason.

Oscar Wilde.

However, a different analysis might lead to the conclusion, or feeling, that reason and emotions are not in intrinsic conflict. Rather, it might be the case that rational thought is ultimately derived from or has to take appropriate account of emotional issues, along with, or maybe even in preference to, cognitive ones. In fact, for many of the tasks that intelligent computers may carry out it is hard to imagine how they might perform them satisfactorily without some emotional component in their performance. For example, if a program performing medical diagnosis is to interact with patients then it will surely be required to indicate some sympathy with them if it is to be acceptable.

The interplay between emotions and thoughts is indicated by the description in Antonio Damasio's influential book *Descartes' Error*:

I see the essence of emotion as the collection of changes in body state that are induced in myriad organs by nerve cell terminals, under the control of a dedicated brain system, which is responding to the content of thoughts relative to a particular entity or event.

Antonio Damasio (1994), Descartes' Error: Emotion, Reason, and the Human Brain, New York: Putnam.

Damasio was by no means the first to dispute Descartes' separation of mind and body but, in arguing that the mind interacted with the rest of the body, he went further in suggesting that 'dispositional representations' activate 'somatic markers' that function as emotional states that permeate all other activities, determining, for example, their relative priority. This is a theory of neuroscience and, as such, has no relevance to AI unless it is found necessary to emotionalise computers, it is decided to replicate humanoid functions to achieve this, and the theory is considered to be sound.

The proposal to endow machines with emotional capabilities, or at least to accept that it is reasonable to attribute emotions to them, is of some vintage:

I now believe that it is possible to construct a supercomputer so as to make it wholly unreasonable to deny that it had feelings.

Michael Scriven (1953), The mechanical concept of mind, Mind, 61, 320-340, reprinted in Alan Ross Anderson, ed. (1964), Mind and Machines, Englewood Cliffs, N.J.: Prentice-Hall.

Such a proposal overlooks the moral of Karel Capek's play *R.U.R. (Rossum's Universal Robots)* of 1920, which introduced the word 'robot' (from the Czech word 'robota', meaning servitude) to the English language. Actually, Capek had used 'robot' in a short story written in 1917 but it is the 1920 play, translated into English in 1923, which became famous. In *R.U.R.* the robots at first had no emotions and were perfectly happy, if that were possible, in their role as slaves. Only when an irresponsible scientist endowed them with feelings did they become so frustrated and angry at their treatment that they rebelled and killed all human beings.

In a parallel argument to that we have seen concerning intelligence, one can argue about whether an indication of, say, happiness or sympathy is really a simulation or pretence of an emotional attitude, rather than a genuine one. Even if there is no technical basis for such an attribution to computers it may be helpful, just as saying that a rumbling volcano is angry encourages people to move away. If an indication is acceptably convincing then we, as observers of a program, might well attribute emotions to it, regardless of how that indication has been generated:

> **We shall not be able to write programs for computers that allow them to respond flexibly to a variety of demands, some with real-time priorities, without thereby creating a system that, in a human, we would say exhibited emotion.**
>
> *Herbert Simon (1977), Models of Discovery: and other topics in the methods of science, Boston, Mass.: D. Reidel Publishing Company.*

But just as users of the ELIZA program may be only too willing to ascribe understanding to it, so users of other programs may too easily ascribe emotions to them.

Also, users of computers, as of many other machines, are very easily led to adopt emotional attitudes towards those computers:

> **Dirksen pressed her lips together tightly, raised the hammer for a final blow. But as she started to bring it down there came from within the beast a sound, a soft crying wail that rose and fell like a baby whimpering. Dirksen dropped the hammer and stepped back, her eyes on the blood-red pool of lubricating fluid forming on the table beneath the creature. She looked at Hunt, horrified. "It's … it's – ."**
>
> **"Just a machine," Hunt said, seriously now.**
>
> *Terrel Miedaner (1977), The Soul of Anna Klane, Church of Physical Theology, Ltd.*

Miedaner's speculations on the theme of how our emotional attitudes to machines change with our perceptions of their lifelikeness have gained currency through the development of virtual reality games that may habituate

behaviour towards artificial, but life-like, animals (including humans) so that may it carry over into interactions with real animals.

We might also imagine designing computers that attempt to imbue users with appropriate emotions. In this case perhaps designers will rehabilitate the Doctrine of the Affections, which provides a set of rules relating musical constructs to emotions, for example, a rapidly rising sequence of thirds was considered to induce euphoria and a lamento bass to cause sadness. This doctrine was first described by Athanasius Kircher in his *Musurgia Universalis* of 1650. It also contained details of the remarkably useful arca musarithmica, a mechanical, water-driven device to compose music, as well as write messages in cipher, calculate the date of Easter in any year, and design fortifications.

However, more under consideration in this section is the computer's own understanding of and having of emotions rather than *our* attitudes to computers. As with language understanding and generation, we need to be careful to distinguish issues concerned with computers recognising emotions in their users and with computers having emotions themselves.

Regarding emotion recognition, it is possible to use sensors to monitor various physiological states, such as body temperature, blood pressure, muscular tension, sweating, and so on, and to have devices detect, for example, pressure on the keyboard, facial expression and eye dilation. Is it possible for a computer to use such information to determine emotional states? Rosalind Picard, Professor of Media Technology at MIT, is optimistic:

> **Despite its immense difficulty, emotion recognition is easier than thought recognition ... largely because there are not as many emotions as thoughts.**
>
> *Rosalind Picard (1997), Affective Computing, Cambridge, Mass.: MIT Press.*

It is doubtful that thoughts and emotions are countable, but perhaps the suggestion is that there are an infinite number of thoughts and only a finite number of emotions. If so, they are manifested in a multitude of ways, judging by a report on the work of Javier Movellan of the Institute for Neural Computation at the University of California San Diego, whose devices can recognise from the face alone hundreds, and eventually millions, of expressions:

> **Movellan's devices now can identify hundreds of ways faces show joy, anger, sadness and other emotions. The computers, which operate by recognizing patterns learned from a multitude of images, eventually will be able to detect millions of expressions.**
>
> *Charles Piller (2002), A human touch for machines, Los Angeles Times, May 7.*

The devices have learned to categorise facial expressions after analysing thousands of videotapes described in terms of a set of 44 discrete 'action units' that form facial muscle movements itemised by the psychologist Paul Ekman in the 1970s.

It is usual to distinguish between primary and secondary emotions. The former are seemingly automatic, instantaneous responses to situations, for example, being alarmed, nauseated or aroused, that, because they involved physiological changes that have evolved to be beneficial, sensors could probably detect. The latter are states that are generated by cognitive processes involving some reflection on past, present or future situations, for example, being anxious, excited or embarrassed. Sometimes, secondary emotions give rise to physiological changes – but not always. Therefore, the detection of secondary emotions through sensors seems unlikely. Even if the nature of the emotion could be identified, a computer would probably not be able to determine what gave rise to it.

The primary-secondary distinction with emotions bears comparison with the reactive-deliberative distinction with cognitions. It is always arguable whether a particular response is an automatic reaction or a result of deliberation, whether conscious or not. Is love at first sight possible? Does counting to ten prevent anger? At least, it is clear that the study of affective computing is at an early stage.

Little account is taken of the fact that emotions such as anxiety usually arise not in response to a particular entity or event (as Damasio says in the quotation above) but in response to an aggregation of entities or events. Moreover, we rarely can be said to be in a single emotional state – we are in many of them much of the time, to a greater or lesser extent. You probably have mixed emotions about the prospect of computers monitoring your innermost feelings and empathising with them. Whether sensors can disentangle these remains to be seen.

The use of sensors to detect emotional states is part of a broader move towards 'wearable computers'. Bringing a new meaning to the phrase 'smart clothes', researchers envisage burdening the human body (already laden with wristwatch, mobile phone, pager, calculator, organiser, and so on) with mobile multimedia, wireless communication, and wearable technology, enabling the wearer to interact more conveniently with others in modern society:

> **Smart glasses, smart shoes …, and smart undergarments electronically sense, for example, my heart rate, skin resistance, and body temperature. Should someone pull out a gun and demand my money, my smart clothing**

> **might respond appropriately … by virtue of the sudden increase in heart rate without any increase in physical exertion.**
>
> *Steve Mann (1996), Smart clothing: the shift to wearable computing, Communications of the ACM, 39, 8, 23-24.*

In such a circumstance, the undergarments might not remain smart. Perhaps that is not what is meant by an appropriate response.

This technology was used in the first 'emotional car', the Pod, unveiled at the Tokyo Motor Show in 2001. If sensors find that the driver's pulse rate and level of perspiration are rising then the car advises the driver to calm down, plays soothing music and blows cool air into the car. When the car nears a restaurant that serves the driver's favourite food, it informs the driver, presumably, after first detecting hunger pangs. In addition to attempting to detect the driver's state, the Pod manifests emotional behaviour of its own. If the driver has been away for a while, the Pod says that it has missed them. The front of the car changes colour to show its feelings – orange-yellow for 'happy' (when the driver approaches), blue for 'sad' (when it runs out of petrol) and red for 'angry' (when the driver brakes too hard). Later versions will turn green with envy when a more stylish car passes.

These superficial gestures of emotion make it clear that the input-output nature of emotions is very different to that of language, which exists only in an externalised form. Emotions are to a large extent internalised. Just as it is hard for a computer to detect if we are anxious, so it would be hard for us to tell if a computer were anxious, assuming that it were possible for a computer to be in such an emotional state, unless it used gauche signals like the Pod unlike the subtle, but easily misunderstood, human ones.

Meanwhile, until computers have got a better grip on their and our emotions, it might be better for them to be rather unemotional:

> **It would probably be dangerous to give early generations of universal robots a capacity for anger, which might be triggered by a misunderstanding in their limited view of the world. Later robot generations may occasionally find some kind of anger useful in interactions with irresponsible humans or irresponsible robots.**
>
> *Hans Moravec (1999), Robot: Mere Machine to Transcendent Mind, Oxford: Oxford University Press.*

This restraint with respect to emotional capabilities is notable coming from Hans Moravec, director of the Robotics Institute at Carnegie Mellon University, who, as we will see, has predicted the imminent arrival of robots with every other capability.

41. Agents: "wrong and evil"

Work on computational emotion has been given a recent impetus by research on 'lifelike agents', which are a specific application of the ubiquitous notion of an 'agent'. As in the everyday sense of an agent, a computational agent is intended to act on behalf of someone to carry out some task. To be useful, an agent should have some initiative, autonomy and knowledge in order to be able to carry out its task without requiring explicit specification by the user of all the steps required. In short, an agent needs intelligence.

Research on agents exploded in the 1990s, with almost any advanced computational project being described as 'agent-based'. There were two main reasons for this. First, agent-based projects enabled AIers to continue their research without having to call it AI, which had become a term of disrepute following the perceived failure of the expert system industry. The notion of a knowledgeable, autonomous agent brings with it all the long-standing issues of AI (planning, communicating, reasoning, learning, and so on), together with some new ones, such as emotion. The fact that the long-standing AI issues were still standing did not diminish the enthusiasm for the prospect of agents.

Secondly, it was becoming more apparent that ordinary users were having increasing difficulty in coping with the complexities of contemporary computer systems. For example, the World Wide Web provided access to billions of documents but users could not easily locate what they needed. Wouldn't it be nice if users could call upon an agent that would search on the user's behalf, taking account of the user's individual needs? In general, the idea is that the agent would work autonomously on behalf of the user, that is, without requiring the user to enter a command whenever a task is to be carried out.

Nicholas Negroponte, co-founder and chairman of the MIT Media Laboratory and author of the best-selling *Being Digital* (1995), found a metaphor for an agent:

> **The best metaphor I can conceive of for a human-computer interface is that of a well-trained English butler. The 'agent' answers the phone, recognizes the callers, disturbs you when appropriate, and may even tell a white lie on your behalf.**
>
> *Nicholas Negroponte (1997), Agents: from direct manipulation to delegation, in Jeffrey Bradshaw (ed.), Software Agents, Menlo Park: AAAI Press.*

So the leader of the most avant-garde technology research centre proposes the adoption of the anachronistic role of butler as a model for computational

agents. Today, in these egalitarian times, few other than royalty employ butlers but in the days when butlering thrived there was considerable discussion of what the activity entailed:

> **What is a great butler? ... The Hayes Society claimed to admit butlers of 'only the very first rank' [and considered that] 'the most crucial criterion is that the applicant be possessed of a dignity in keeping with his position' ... If one looks at these persons we agree are 'great' butlers ... it does seem to me that the factor which distinguishes them from those butlers which are merely extremely competent is most closely captured by this word 'dignity' ... It is sometimes said that butlers only truly exist in England. Other countries, whatever title is actually used, have only manservants ... Continentals are unable to be butlers because they are as a breed incapable of the emotional restraint which only the English race is capable of.**
>
> *Kazuo Ishiguro (1989), The Remains of the Day, London: Faber and Faber.*

The Japanese-born Ishiguro moved to England when he was five, so I suppose we must allow him to pontificate on what the English race is capable of. The multitudinous properties (discussed below) that have been put forward to characterise computational agents have not yet included that of dignity, notwithstanding the fact that Rosalind Picard gave a talk on "Human and Machine Dignity" at a conference on "The Implications of Artificial Intelligence Upon Jewish and Christian Understandings of Personhood" at MIT in 1998.

So, agents are, in Ishiguro's words, "merely extremely competent", if we are very lucky. We can also see that, although agent-based research has been used to justify efforts to assign emotional properties to computational devices, computational butlers might be better characterised by their emotional restraint. Or perhaps it is necessary for computers to have a deep understanding of emotion in order to be able to restrain it with dignity? Anyway, it seems that the woefully sexist term of 'manservant' might be more appropriate than that of 'butler' as a metaphor for an agent.

The eagerness with which the term 'agent' has been employed makes it hard to find a consensus definition that will cover all its uses. However, we can list some of the attributes that agents are intended to have:

- **Reactivity: the ability to selectively sense and act**
- **Autonomy: goal-directedness, proactive and self-starting behavior**
- **Collaborative behavior: can work [with others] to achieve a common goal**
- **Communication ability: [uses] language resembling human-like 'speech acts' …**
- **Inferential capability: …**

- **Temporal continuity: persistence of identity and state …**
- **Personality: [manifests] attributes of a 'believable' character …**
- **Adaptivity: being able to learn …**
- **Mobility: being able to migrate ... from one host platform to another**

Jeffrey Bradshaw (1997), An introduction to software agents, in Jeffrey Bradshaw (ed.), Software Agents, Menlo Park: AAAI Press.

Not all so-called agents would have all these properties but we would expect them to have a core subset (say, reactivity, autonomy, temporal continuity, and inferential capability) plus some of the others.

It is useful to separate these attributes into those that are internal (concerned with how the agent works) and those that are external (concerned with how the agent behaves and looks). In the former case, as a software concept, an agent can be seen as an extension of the established line of AI and programming research. The basic idea is not so different from that underlying the Advice Taker project described by John McCarthy in the 1950s. The Advice Taker would receive its input in natural language and then autonomously deduce implications from the facts it already knew and take appropriate decisions.

An agent is similar to an object (the software concept discussed earlier). One of the earliest uses of the object concept occurred under the name of 'actor' in the programming language devised by Carl Hewitt that was used by Terry Winograd in the SHRDLU project:

[An actor] is a computational agent which has a mail address and a behavior. Actors communicate by message-passing and carry out their actions concurrently.

Carl Hewitt (1977), Control structures as patterns of passing messages, Artificial Intelligence, 8, 323-363.

An actor's a guy who, if you ain't talking about him, ain't listening.

Marlon Brando (1956), quoted in Bob Thomas (1973), Brando, as frequently ascribed to Brando, although he probably heard it from the film producer, George Glass.

Actually, Brando had it exactly wrong: an actor (or object or agent) is a software entity that is listening all the time, whether or not any messages are directed towards it. When an incoming message is relevant, an object responds to it, by communicating indirectly with other objects by broadcasting messages. Objects therefore have, from the above list, the properties of reactivity and temporal continuity. Agents go beyond objects in having an associated 'mental state' which can be viewed as consisting of various cognitive notions. An agent computation involves processing these mental components in ways similar to conventional AI.

The fact that something (A) can be viewed as consisting of something (B) does not necessarily mean that A actually has B. The point is that an agent-oriented user or programmer finds it useful to ascribe mental properties to the software, even if it is not strictly legitimate:

> **To ascribe certain *beliefs, free will, intentions, consciousness, abilities* or *wants* to a machine or computer program is *legitimate* when such an ascription expresses the same information about the machine that it expresses about a person. It is *useful* when the ascription helps us to understand the structure of the machine, its past or future behavior, or how to repair or improve it... Ascription of mental qualities is *most straightforward* for machines of known structure such as thermostats and computer operating systems, but it is *most useful* when applied to entities whose structure is very incompletely known.**
>
> *John McCarthy (1979), Ascribing mental qualities to machines, Technical Report, Memo 326, AI Lab, Stanford University.*
>
> **An agent is an entity whose state is viewed as consisting of mental components such as beliefs, capabilities, choices, and commitments. These components are defined in a precise fashion, and stand in rough correspondence to their common sense counterparts. In this view, therefore, agenthood is in the mind of the programmer.**
>
> *Yoav Shoham (1993), Agent-oriented programming, Artificial Intelligence, 60, 51-92.*

This, then, is an attempt to avoid getting embroiled in philosophical debates about whether agents *really* have, in the sense of physically possess, various properties. McCarthy emphasises the point of view of an observer finding it useful to ascribe mental qualities to a machine, as indeed an observer is inclined to do if the machine's functioning is not fully understood. Shoham is more concerned with the fact that it is helpful to the system designer to view the system in terms of agents. In both cases, just as beauty is in the eye of the beholder, so agency is in the mind of the observer – if he or she says it's an agent, nobody can deny it. Is this any different to Australian aborigines ascribing properties, such as jealousy and anger, to things in their environment, such as clouds and rocks? The intention, in both cases, is that such ascriptions help decision-making about the system being considered.

This is a potential point of confusion that we see throughout AI. When it is said that an agent, human or computational, has a particular attribute this might be a case of an observer finding such an attribution useful (in order to be able talk about the agent's behaviour or to make predictions about what the agent might do), without claiming that the agent physically possesses that attribute in some sense:

> **People find it useful to attribute beliefs to others … It is important to note that it is not too relevant whether or not beliefs have any kind of real existence somewhere in our minds (whatever that might mean). We observe merely that our reasoning processes seem to make use of these abstractions.**
>
> *Michael Genesereth and Nils Nilsson (1987), Logical Foundations of Artificial Intelligence, Los Altos, Ca.: Kaufmann.*

Of course, the claim that some entity exists within some agent is much harder to substantiate in the case of humans than it is for computers, whose innards have been designed and may be inspected by us.

Others, however, when they make such an attribution intend it to be understood that there is within the agent something which can be identified with that attribute. For a computational agent, there would be some symbolic representation of a belief, for example, and for a human agent, some analogous symbolic structure in the brain.

This distinction is fairly clear for cognitive attributes such as beliefs but it is less clear for an attribute such as pain. If we see a human walk into a lamppost we can imagine the internal physical changes corresponding to the display of pain observed. We can also imagine that it might be possible to program a robot to display all the external manifestations of pain if it bumped into a lamppost. We might also imagine that, in so far as we understand the human physical changes, there might be – in fact, might have to be if the external manifestation is convincing – corresponding changes within the robot's internal structures. And yet, however deeply we philosophise about this, we still feel that the robot can only be simulating pain, which necessarily has a biological basis. Some people feel the same way about beliefs and all other attributions.

Re-launching AI as research on intelligent agents does not diminish its propensity to provoke hostile reactions, for the proposition that agenthood exists only in the imagination of the user (and that there may or may not be any justification for this attribution) may itself be considered dangerous if it diminishes our own responsibility and lowers our own self-worth:

> **The idea of 'intelligent agents' is both wrong and evil … Evil, because they make people diminish themselves, and wrong, because they confuse the feedback that leads to good design.**
>
> *Jaron Lanier (1995), Agents of alienation, Journal of Consciousness Studies, 2, 76-81.*

Such opinions are passionately held, and the more passionately so, the more incoherent they seem to be. It is hard for AIers to engage with critics with such prejudices, which are not based on any understanding of what is actually

being done, at a technical level. Jaron Lanier, computer scientist, composer and artist, is well known for coining the term 'virtual reality' and as a provocative speaker on technology, philosophy and politics, so much so that a postage stamp has been issued in his honour by the country of Palau, no less. However, like many critics of AI, he has not worked in the field of AI itself. His criticism seems no more profound than an argument that it is dangerous to attribute beliefs ("He thinks he's hungry") and intentions ("He wants to go for a walk") to a dog. Perhaps unfortunately, AIers press on regardless with the development of agents, which in this view is in the mainstream of AI, focussing on the representation and processing of cognitive components.

42. Collaboration: "no man is an island"

The singular view of an agent, emphasising its mental state, does not take account of one of the attributes desired of agents mentioned above, that they be able to collaborate with others. The properties of a set of collaborating agents are not just the union of their individual properties:

> **Collaborative plans cannot be recast simply in terms of the plans of individual agents, but require an integrated treatment of the beliefs and intentions of the different agents involved … Thus, capabilities for collaboration cannot be patched on, but must be designed in from the start.**
>
> *Barbara Grosz and Sarit Kraus (1996), Collaborative plans for complex group action, Artificial Intelligence, 86, 269-357.*
>
> **No man is an island, entire of itself; every man is a piece of the continent.**
>
> *John Donne (1624), Devotions.*

This aspect of agent research merges with the earlier stream of work on 'distributed AI', which is concerned with developing techniques whereby different components may cooperate through negotiation and planning and with designing computational architectures suitable for distributed computations in applications such as computer-aided design and computer-supported collaborative work. Indeed, it is interesting that developers of distributed, networked systems, who were without an interest in AI, had independently found knowledge ascription useful, for they talked in terms of a node knowing that another node had sent a message to a third node but not knowing whether it had been received and acknowledged, and so on, and subsequently came to adopt the theoretical models of knowledge developed in AI.

The emphasis in distributed AI is slightly different to that in multi-agent systems. In distributed AI, components are designed to cooperate through

task sharing and negotiation, in order to work together coherently to solve some problem. In multi-agent systems, an agent is designed to operate autonomously and may cooperate with other agents only if it considers that such cooperation may be beneficial to it. There still needs to be a concern for establishing bases for cooperation and coherent interactions but there may not necessarily be a focus on a common global objective.

This is all part of a trend in AI and computing from the individual to the social. Early work in AI focussed on developing systems to perform isolated problem solving. AI was not alone in this focus. Epistemology, for example, has been almost exclusively concerned with the knowledge of a single agent. It has not really considered what it might mean for a group of agents to know something or taken account of the fact that for many activities, such as diplomacy or trade, it is necessary for agents to consider the knowledge, beliefs, desires, and so on of the other agents involved. Therefore, the development of collaborative agents began in something of a theoretical vacuum.

The trend to socialise AI has arisen for several reasons. First, the notion of intelligence is itself a social construct and many of its aspects, for example, the use of language, only make sense within a social context. According to the social development theory of Lev Vygotsky (1896-1934), social functions are prior to individual ones:

> **Every function in the child's cultural development appears twice: first, on the social level, and later, on the individual level; first between people (interpsychological) and then inside the child (intrapsychological).**
>
> *Lev Vygotsky (1978), Mind in Society: the development of higher psychological processes (edited and translated by Michael Cole et al), Cambridge, Mass.: Harvard University Press.*

Perhaps computer functions need to develop in the same way. Secondly, it has been increasingly argued that mental skills mainly arise from the agent (human or system) being embodied within a context, so that behaviour-based robots are needed to provide the fundamental basis for machine intelligence. Also, of course, we now want to embed intelligent systems in the environment to support human activities and hence to be able to engage in social interactions with humans and other systems:

> **In fact social intelligence is one of the ways AI reacted to and got out of its crisis … In the 1960s and 1970s this gave rise to cognitive science, now it will strongly impact on the social sciences.**
>
> *Cristiano Castelfranchi (1998), Modelling social action for AI agents, Artificial Intelligence, 103, 157-182.*

It is not coincidental that multi-agent systems research grew just as the technology of networked computers made social factors important.

43. Animation: "human-like characteristics"

The other main view of agents, that which focuses on the external aspects, that is, on how the agent is portrayed to the user, derives from the problems inherent in the process of delegation involved in the user-agent relationship. As with other automatic aids, such as autopilots and expert systems, users need to be given ways of developing trust in and sharing responsibility with agents. For example, REA (an 'embodied conversational agent' playing the role of a real estate agent) communicates via speech, facial expressions and gestures, recognised and synthesised.

It is argued that agents need to be given life-like attributes so that we may better relate to them. Anthropomorphising agents may have benefits in bringing into play many natural interactive techniques and conventions that we have developed for communicating with individuals in societies:

> **We surmise that once people are accustomed to synthesized faces, performance becomes more efficient, and a long partnership further improves performance. Human-like characterization is one good form of autonomous agents, because people are accustomed to interact with other humans.**
>
> *Akikazu Takeuchi and Taketo Naito (1995), Situated facial displays: towards social interaction, Proceedings of the Conference on Human Factors in Computer Systems, New York: ACM Press.*

This, of course, is a very superficial argument. A great many conventions for human-human interaction have developed – for example, in most cultures we greet one another by shaking hands, for reasons anthropologists could explain – that we would not expect to transpose to human-computer interactions. The issue remains of which 'human-like' techniques are useful for human-computer interaction.

If it turns out that computer agents are incapable of sustaining the anthropomorphised façade that we give them then the loss of face may have a significant negative impact on users. We all know the effect that a misplaced smile, grimace or wink may have. Is there any chance that we may imbue computers with sufficient understanding of the subtle nuances involved in order to be able to use such gestures reliably? If these gestures were missing from a human-like computer-generated face, would this cause more problems than not having a facial image at all?

Also, of course, it is not just a matter of generating such signals – the computer needs to be able to recognise similar signals on the human face in order to be able to determine a suitable response:

When he appears as a Ghost he had
A countenance more in sorrow than in anger.

William Shakespeare, Hamlet, 1, 2.

If we succeed in humanizing the computer interface in one way, say, in the use of facial gestures, then will this not encourage the users to over-generalise and assume that the agent also has other human-like properties:

Don't look at me, Sir, with – ah – in that tone of voice, Sir!

Punch (1884), 38.

For example, we might also reasonably expect the computer to be able to generate the right tone of voice, corresponding to sympathy, anger, frustration, impatience, and so on.

It may also be the case that the human metaphor is too limiting. There are many activities where we do not restrict computers to what humans can do. For example, we accept that music does not have to be limited to what the human voice can produce, and we are coming to accept that, through computer-synthesised music, it does not need to be restricted to what can be played by humans with two hands of a limited finger range on non-electronic instruments. It is clearly the case that computers can, in some respects, go beyond human capabilities, so why encourage a human metaphor that may unnecessarily limit users' expectations?:

We should have much greater ambition than to make a computer behave like an intelligent butler or other human agent. Computer-supported cooperative work, hypertext-hypermedia, multimedia, information visualization, and virtual reality are powerful technologies that enable human users to accomplish tasks that no human has ever done. If we describe computers in human terms, we run the risk of limiting our ambition and creativity in the design of future computer capabilities.

Ben Shneiderman (1997), Direct manipulation versus agents, in Jeffrey Bradshaw (ed.), Software Agents, Menlo Park: AAAI Press.

This is part of a widespread argument in computing that its goal should not be artificial intelligence but 'augmented intelligence': we should aim not to replicate human intelligence but to help users extend their own intelligence.

Even if it is felt necessary to provide an agent with an artificial personality, it does not follow that it has to be humanoid. Indeed, a menagerie of supposedly life-like animated agents now exists. For example, one of the earliest animated agents was the Personal Digital Parrot One (PDP1), parrots being renowned for their advanced linguistic abilities and broad range of

emotions, as acknowledged by the expression "sick as a parrot" and the Monty Python dead parrot sketch. An alliance with the film animation industry looms:

> **You can hardly tell where the computer models finish and the real dinosaurs begin.**
>
> *Laura Dern, commenting on the film Jurassic Park.*

However, it may be that the addition of emotion will not improve the animated reality already achieved.

With the suspicion that the research is primarily great fun, it is reasonable to ask if there is any evidence to support the conjecture that life-like, animated agents improve human-computer interactions. It would seem to be a conjecture open to empirical investigation. Unfortunately, there are so many variations in the features of agents, the tasks to be performed, the users involved, the measures of performance, and so on, that it is hard to derive definite conclusions. All that can be said is:

> **To date, the literature does not provide evidence for a so-called persona effect, that is, a general advantage of an interface with an animated agent over one without an animated agent.**
>
> *Doris Dehn and Susanne van Mulken (2000), The impact of animated interface agents: a review of empirical research, International Journal of Human-Computer Studies, 52, 1-22.*

Nonetheless, the notion of 'agent' continues to flourish in AI even though its adoption for multifarious purposes makes it difficult to pin down its precise contribution.

One factor is the agent's apparent autonomy, which enables it to decide for itself what to do when. This helps in the decomposition of a complex problem and the sharing of control. The fact that an agent determines for itself, based on its awareness of its environment, when to act and how to change its state may reduce programming difficulties. Interactions between agents are determined by the agents themselves and thus can appear to deal spontaneously with unanticipated events, without being the direct concern of the system designer.

Thus, decision-making is more localised and robust, without the need for some kind of overall organisation. On the downside is the possible unpredictability of the system because of the complexity of the interactions between agents and because behaviour may well emerge from the group of agents that is not foreseeable from the properties of the individual agents.

Perhaps the most popular current AI textbook, which sells itself as "the intelligent agent book", dispenses with the difficulty of defining precisely what an agent is:

An agent is just something that perceives and acts.

Stuart Russell and Peter Norvig (1995), Artificial Intelligence: A Modern Approach, Englewood Cliffs, N.J.: Prentice-Hall.

That, at least, has the merit of being brief. It seems to avoid the necessity of dealing with the messy business of 'thinking', at the price of discussing how computers may 'perceive' and 'act'.

44. Perception: "a more searching vision"

The focus on perceiving and acting, which corresponds to an emphasis on the environment rather than the task, is related to the debate on the situatedness or otherwise of intelligence. The agent is considered to have a physical presence or 'body' capable of sensing and affecting the environment, with an awareness of place and time. If information is available through the senses then, so the argument goes, it may not need to be represented internally. Thus such agents are liberated from the need to maintain objectively correct models of the world but may manage with subjectively useful impressions of the world.

Perception is the coming to have knowledge of the environment through the senses, of which there are five for humans: tasting, smelling, feeling, hearing and seeing. Considering these in turn, we can skip the first two as current AI systems lack taste and smell. Feeling, in the sense of touching, is provided by sensors that detect distortions in the objects contacted and provide a 'mental image' of those objects. The engineering technicalities differ according to the degree of anthropomorphism required. The sensor may be part of a humanoid hand required to pick things up; it may be part of an arm with the same segments and degrees of freedom as the human arm; the agent may be required to have human-like proprioceptive properties, that is, to know where the sensors are (humans are fairly accurate in this but nowhere near as good as a robot could be). In some contexts, such as planetary exploration robots, it may be pointless to have sensors that simulate those of humans.

As far as hearing is concerned, in AI the only thing considered to be worth hearing is human speech. Speech is the main mode of direct communication between humans and hence would seem likely to be important for human-computer communication. Speech recognition involves translating analogue signals into digital encodings that are mapped onto words (at least, it is assumed that to understand the signals they need to be converted to words). Speech recognition provides another interesting AI case study. In 1971 the United States set out a large-scale five-year programme

on speech recognition research in which the main projects were AI-oriented, adopting an expert system approach. Some doubts about the feasibility and ethicality of this research were expressed:

> **Granted that a speech-recognition machine is bound to be enormously expensive, and that only governments and possibly a very few very large corporations will therefore be able to afford it, what will they use it for ... surveillance of telephone conversations.**
>
> *Joseph Weizenbaum (1976), Computer Power and Human Reason, San Francisco: W.H. Freeman and Co.*

A critical eye on new technologies is always healthy and warnings of potential abuse are to be welcomed. Today, however, we know that practical speech recognition is not "enormously expensive". Usable systems are available for a few dollars for ordinary computers and many socially acceptable uses have been found: to enable disabled people to use computers; for training systems (for learning languages or in contexts, such as air traffic control, where speech communication is necessary); for straightforward telephone communications; for languages such as Chinese with too many characters for a keyboard. Over 15 million speech recognition chips had been sold by 2002 (over 100 million if we include voice dialling cell-phones). We need therefore to be wary of prohibiting research because first impressions are that applications are likely to be undesirable.

Technologically, however, Weizenbaum's misgivings were justified because it turned out that practical speech recognition has not been feasible with symbol-processing AI. Instead, techniques based on hidden Markov models were found to be much more effective, to the chagrin of AIers. The basic idea of a hidden Markov model can be gained by recalling the Turing machine, in which input symbols drove the machine through new states and generated output symbols. In a hidden Markov model, the state transitions and output generations are probabilistic, so that the state sequence cannot be inferred from the input symbols. Hidden Markov model algorithms determine the state sequences (for example, phonemes) most likely to have generated a particular output sequence (for example, an acoustic signal).

For speech to provide natural human-computer communication it is necessary for there also to be speech output. Hearing computer speech can be less intrusive and more reliable than reading text popping up in a window on the screen. The dream of speaking machines predates AI:

> **The talking machine of Kempelen is not very loquacious but it pronounces certain childish words very nicely.**
>
> *Johann Wolfgang von Goethe (1778), referring to the talking machine built by Baron Wolfgang von Kempelen.*

But the speech generation systems that we are familiar with today make no use of AI techniques. If the context requires it, AI may have a role in determining what needs to be said and maybe in adding some nuances of speech, such as inflections. Current speech synthesisers are adequate for tasks such as enabling blind students to receive computer output.

The final sense, seeing, is the one most associated with the idea of computers perceiving their environment. Computational vision is a vast and complicated field, with techniques from biology, engineering and mathematics as well as AI. Initially, like computational language, it was considered to be a core part of AI but its special techniques soon became divorced from mainstream AI. Only recently, with the growing importance of integrating the various intelligent faculties, has this begun to change.

Human vision is deceptively facile, for most of us – or it seems to be if we give it only superficial thought:

> **Either the human being must suffer and struggle as the price of a more searching vision, or his gaze must be shallow and without intellectual revelation.**
>
> *Thomas de Quincey (1845), Vision of life, Suspiria de Profundis.*

Computational vision seems to be very difficult, for various reasons. First, there is a lot of raw data in visual images, as we are made aware by their transmission times over the internet. The analysis of moving images in real-time, as some projected applications require, is therefore a major challenge. Also, images, which are usually two-dimensional views of a three-dimensional reality, may be ambiguous and unclear. For example, tractors may be confused with tanks. The task of computational vision has been taken to be that of deriving a description of the 3D reality corresponding to the 2D image, although recently it has been argued that this may be unnecessary for understanding and reacting to the image, an argument echoing similar ones doubting the need for detailed analyses of language and planning.

Typically, an attempt is made to reduce the huge amount of data to manageable proportions by detecting edges (corresponding to changes in intensity occurring over a surface boundary) and then aggregating labelled edges into segmented bodies (such as a table, a desk lamp, a tank, a bridge, and so on). The earliest such work considered that the edges in a scene of trihedral objects may be labelled convex, concave or occluding and that, in the real world, only certain combinations of these edges (in fact, 16 of the 64 theoretically possible) can occur at junctions. With the constraint that an edge must have the same label along its length, a systematic search may find a consistent labelling for all edges in the scene.

This search method is now called the Waltz algorithm, in honour of David Waltz, who refined the method in 1975, and in neglect of Max Clowes and David Huffman, who had earlier proposed the method, and of numerous later researchers, who have generalised it to arbitrary polyhedrons and smooth curved objects. The Waltz algorithm is one of the first examples in AI of 'constraint satisfaction'. Many problems can be expressed in terms of a set of constraints or conditions that have to be satisfied and several special-purpose algorithms for CSPs (constraint satisfaction problems) have been developed.

An indication of the complexity of the computational vision field can perhaps be given by listing some of its sub-fields: imaging sensors; radiometry; multispectral sensing; confocal microscopy; signal representation; image warping; edge detection; line labelling; texture analysis; motion analysis; 2½D sketch; stereopsis; segmentation; object recognition; fuzzy image processing; SIMD signal processing; active vision systems; industrial imaging; character recognition; digital mapping; face recognition; image retrieval; image processing applications (for example, to genome topology and plant leaf growth). That's a lot of technicality 'just' to enable a computer to see:

Worth seeing, yes; but not worth going to see.

Samuel Johnson (1779), quoted in James Boswell, The Life of Samuel Johnson (1791), referring to the Giant's Causeway.

So perhaps we may be excused if we leave it at that.

Apart from the extremely technical problems that arise with computational vision, there are psychological and philosophical issues to consider. For example, the standard AI approach, which is to represent the image digitally and then 'parse' it in order to extract a description of its meaning, represented by the actual 3D scene imaged, is fundamentally questionable. It seems to be the case that in some contexts we 'reason visually', that is, by apparently directly manipulating a mental image, rather than by processing any symbolic representation of it. This is indicated by introspecting on how we answer questions such as "How many doors does your house have?" and "What colour is Marge Simpson's hair?". Experiments asking people to say whether two images are the same but in a different rotation show that the reaction time is proportional to the degree of rotation needed. So, there are many open questions about the kinds of visual capabilities that an intelligent agent needs and about how these capabilities may be provided by computational means.

As far as an agent's acting is concerned, this is taken to mean the use of effectors (wheels, hands, legs, car paint sprayers, and so on) that change the

position or other properties of the agent and objects in the environment. As such, it is more a matter of mechanical engineering than AI. Without wishing to denigrate mechanical engineers, it seems, following an in-depth knowledge acquisition consultation with experts in the field, that acting is not so difficult. According to them, acting is:

> **… the most minor of gifts – after all, Shirley Temple could do it at the age of four … just one big bag of tricks … like roller-skating – once you know how to do it, it is neither stimulating nor exciting … standing up naked and turning around very slowly … a matter of calculated instinct … the expression of a neurotic impulse.**
>
> *Katharine Hepburn, Lord Olivier, George Sanders, Rosalind Russell, Ernest Borgnine, Marlon Brando, respectively.*

But for robots, as well as actors, physically acting is just the final step of a set of processes. AIers are more concerned with the processes that determine the acts to perform.

In the context of a dynamic environment, this choice is not necessarily correct in a logical sense but it is intended to be the best that the agent can manage in the circumstances. In an 'economic' theory of rational choice, the set of possible actions would be determined and then evaluated by a utility function that assesses the actions' outcomes against the agent's objectives. Such an approach is often based on the 'principle of maximum expected utility', a fundamental theorem of economics developed in 1944 by Morgenstern and von Neumann. In general, however, this two-stage process is not feasible: there are just too many possible actions to determine them all for evaluation. Instead, resource-bounded agents need somehow to take account of the background of their current intentions to focus their reasoning, to make decisions as best they can within the time available, and to monitor the on-going effect of actions being carried out.

The area of bounded rationality has recently come to be seen as intrinsic to AI, rather than an unfortunate necessity (although Herbert Simon had said in 1958 that the aim is "to find the least-cost or best-return decision, net of computational costs"). Because of the difficulty of the problems addressed and the limited computational resources available, theoretical and empirical studies of the possible trade-offs have grown. For example, in many problem solving situations, such as planning in a changing environment, it is desirable to have an 'anytime algorithm' which has the property of being able to provide the best solution it has found so far at any time that it may be interrupted.

Bounded rationality involves a kind of meta-analysis of problem solving processes to determine or estimate the resources (time, space, etc.) that the

processes would consume if they were actually carried out, and then, on that basis, to decide what resources to allocate to which processes, adjusting them, as necessary, as problem solving progresses. As may be seen, it is an area ripe for theoretical investigation and of promise for practical AI. It may be that the standard AI approach of imagining 'perfectly rational' methods and then somehow making them more efficient, as physics might imagine friction-less worlds and then adapt analyses to the reality, is fundamentally misguided. Whereas frictional worlds may just be a refinement of friction-less ones, efficient rationality may be very different from the idealised form. Of course, bounded rationality is what *we* strive for all the time and it may therefore be worth considering what is known about the psychology of human rationality.

45. Computational psychology: "a vacuous theory"

Implicit in the misgivings about a computer's ability to create or emote is a comparison with human capabilities in these respects and, of course, this computer-human comparison runs throughout artificial intelligence work. The development of AI gave rise to a possible new methodology for psychology, for if the psychologist's task is to open up the closed human mind and if AI programs are considered analogous to programs assumed to exist in the human mind, then the psychologist may study the AI programs instead, which are, of course, not closed because they are designed by us:

> **The advent of Artificial Intelligence is the single most important development in the history of psychology ... indeed, it seems to me not unreasonable to expect that Artificial Intelligence will ultimately come to play the role vis a vis the psychological and social sciences that mathematics, from the seventeenth century on, has done for the physical sciences.**
>
> *Alan Allport (1980), Patterns and actions: cognitive mechanisms are content-specific, in Guy Claxton (ed.), Cognitive Psychology: New Directions, London: Routledge and Kegan Paul.*
>
> **The task of a psychologist trying to understand human cognition is analogous to that of a man trying to discover how a computer has been programmed … None of [these AI programs] does even remote justice to the complexity of human mental processes. Unlike men, "artificially intelligent" programs tend to be single minded, undistractable, and unemotional.**
>
> *Ulric Neisser (1967), Cognitive Psychology, New York: Appleton-Century-Crofts.*

Ulric Neisser of Cornell University is considered to have named and defined the new field of cognitive psychology in his 1967 book. For cognitive psychologists, the main interest in AI lies in developing programs that are less single-minded, undistractable and unemotional (and less any other characteristic that computers are considered to have and humans are believed not to), so that the programs might then provide some insight into the corresponding human characteristics. On the face of it, AI seems to be whistling against the wind in this respect, for the trend in psychology is to argue that the proper focus for a study of humankind must be humankind itself, not some piece of machinery. However, the actual machinery is not the point of AI, so possibly AI has a contribution to make to psychology.

All academic subjects have established methodologies for developing their theories. Physics and other natural sciences rely mainly upon mathematical manipulations and experimental studies. Philosophers work mainly with text-based arguments. So did psychologists until it began to use mathematics (but only for very restricted parts of psychological enquiry, such as the learning of paired associates, for example, vocabulary lists) and laboratory-based experimental studies (but only at the cost of placing experimental participants in unnatural situations). Now, specifying psychological theories as computer programs provides a new methodology:

> **Simulation models ... of modern psychology play an epistemic role not unlike the illustrative analogies, case examples, and language-games which Wittgenstein made popular in modern analytical philosophy.**
>
> *Juan Pascual-Leone (1976), Metacognitive problems of constructive cognition: forms of knowing and their psychological mechanism, Canadian Psychological Review, 17, 110-125.*

For example, imagine that a psychologist has a theory about how human memory is organised and that she wishes to apply her theory to explain how a question such as "What is the email address of Julius Caesar?" is answered. Assuming that the answer to such a question is not given instantaneously (it isn't: it is possible to measure the response time), it seems reasonable to suppose that there are some internal, 'cognitive' sub-processes involved in deriving the response. She might seek to define how these sub-processes interact in such a way that they may be expressed as components of a computer program.

She might then compare the processes postulated by her theory for this question with those involved with questions such as "Who killed Julius Caesar?", "Who wrote *Julius Caesar*?", "In which country was Julius Caesar born?", and so on. Her theory might then predict the relative speeds at which the answers to these questions are given. To be clear that her theory does

indeed make such predictions she could express her theory as a program and see how long the program takes to answer the same questions. It is likely that any such theory will be so complicated that she will be unable to derive such predictions without computational assistance. If, now, the predictions accord with experimental measurements of human response times then this is some confirmation that the theory (as expressed in the program) is sound. If the predictions are inaccurate then she could attempt to modify the program, in some theoretically motivated fashion, to improve the correspondence with experimental measurements. Thus, the program becomes the theory: it comes to be regarded as the most useful and precise way of expressing the theory.

In practice, the methodology involves an intricate interplay between experimental studies and program design. For example, measuring the response time of a program with a network representation of knowledge to questions such as "Does an eagle have wings?" and "Does an ostrich fly?" provoked a series of psychological experiments (and then program refinements and then further experiments and so on), leading to the 'spreading activation model', which is a central part of the ACT* theory of human cognition developed by John Anderson. However, unless the theory is tightly defined independently of the program it may be easy to adjust the program to fit any experimental data encountered:

> **We are in danger of having a vacuous theory because we may be able to accommodate any potentially embarrassing result by suitable assumptions about knowledge representation.**
>
> *Mark Singley and John Anderson (1989), The Transfer of Cognitive Skill, Cambridge, Mass.: Harvard University Press.*

ACT* is in fact one of the most detailed of such theories. It assumes that cognitive systems are based on production rules specifying actions that should be carried out when certain conditions hold. The theory distinguishes three stages in cognitive and motor skill acquisition: first, declarative representations are interpreted; second, these representations are compiled into task-specific procedures; third, these procedures are then tuned to become more efficient.

One of the first to follow the methodology of developing a computer program to serve as a psychological theory was Edward Feigenbaum, who, as we have seen, went on to pioneer expert systems, which are explicitly not simulations of human performance. In 1961 he developed a program called EPAM (for Elementary Perceiver and Memorizer, even though the program had no perceptual system) that performed simple verbal learning tasks through discrimination processes. Referring specifically to EPAM, Walter Kintsch gave an idealistic view of the methodology:

> **If the computer learns in the same way as human subjects do in a "real" experiment, we can conclude that the computer program (our theory) is adequate; if the computer does things differently than human subjects, we know that something is wrong in the instructions to the computer.**
>
> *Walter Kintsch (1970), Learning, Memory, and Conceptual Processes, New York: John Wiley & Sons.*

However, we need to be very careful in our comparisons of performance data and in our assumptions of equivalence for internal mechanisms. Conflating the notions of computer program and psychological theory is difficult because a program can be described at many different levels of detail:

> **When an AI model of some cognitive phenomenon is proposed, the model is describable at many different levels, from the most global phenomenological level at which the behaviour is described (with some presumptuousness) in ordinary mentalistic terms down through various levels of implementation, all the way down to the level of program code and even further down, to the level of fundamental hardware operations, if anyone cares. No one supposes that the model maps onto the process of psychology and biology all the way down.**
>
> *Daniel Dennett (1984), Cognitive wheels: the frame problem of AI, in Christopher Hookway (ed.), Minds, Machines, and Evolution, Cambridge, Mass.: Cambridge University Press.*

So, a theorist needs to be clear which level corresponds to her theory.

It is no different to when in 1628 William Harvey proposed his theory of blood circulation in which the heart was described as a pump. (To forestall a flood of corrections, the Chinese had developed such a theory two thousand years earlier, but that doesn't affect the point at issue here.) A theory is at a relatively high level, at which there is some mapping of model features onto what is being modelled. For theories to function as working models they need to be supplemented with implementation details which are not intended to bear any relation to what is being modelled. The intention is that, at the appropriate level, there is some functional equivalence between the theory and what is being modelled. Unfortunately, especially in the case of computer programs, it is hard to be sure that any observed equivalence is not dependent on supposedly irrelevant implementation details and, conversely, that any actual equivalence is not similarly obscured. Also, because of the inscrutability of program code, it may not be clear that it faithfully reflects what is claimed of the theory. Just as William Harvey thought that blood contained a person's soul, so it is possible for a cognitive psychologist to attribute more to a programmed model than is really there.

Computational cognitive theories are intended to demonstrate some functional equivalence to human capability. The equivalence is clearly not at the hardware level but at some level of description of the software. Typically, such theories are expressed in terms of goals, knowledge structures, component processes, and so on:

> **No one is likely to confuse a program embodying a piece of physics with the actual physical process that is being simulated. A program that represents a wave breaking on a shore is manifestly different from a real wave, and it would be absurd to criticize the program on the grounds that it was not wet. No sane person is likely to assume that the real wave is controlled by a computer program: it is governed by physical forces that are simulated by the program. All theories are abstractions, of course, but there is a more intimate relation between a program modelling the mind and the process that is modelled. Functionalism implies that our understanding of the mind will not be further improved by going beyond the level of mental processes. The functional organization of mental processes can be characterized in terms of effective procedures, since the mind's ability to construct working models is a computational process.**
>
> *Phil Johnson-Laird (1983), Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness, Cambridge, Mass.: Cambridge University Press.*

Even if we don't go along completely with Johnson-Laird's assumption that the mind functions as a computational process, we can see that this frequently quoted statement on the nature of computational psychological theories raises difficult questions. Yes, we may think it absurd to imagine a wave simulation program to be wet – because we assume that it is possible adequately to simulate (the appearance of) waveness without addressing wetness. But it is not at all obvious that it is possible to simulate (the appearance of) mental processes such as those involved in reasoning, emotion, pain, and so on, without addressing 'physical' properties. The proponents of behaviour-based AI argue that it is not.

A program developed by Kenneth Colby and colleagues to simulate paranoia indicated the theoretical and applied benefits of this new research methodology. The program interacted via an ELIZA-like interface, maintaining measures of its anger, fear, mistrust, and so on, according to its theory of paranoia. When the transcripts were sent to psychiatrists, none of them realised that they were diagnosing a computer and 23 out of 25 deemed that there were signs of paranoia. In other words, the program might be considered to have a passed a Turing test kind of imitation game for paranoia. Since we hardly need paranoid computers, what use is that?:

A good model of paranoia (in the sense of being a good imitation) would have obvious pedagogical and technological implications for psychiatry. For example, one might subject it to experiments designed to modify its paranoid [input-output] behavior and apply the favorable results to human patients.

*Kenneth Colby, Sylvia Weber and Franklin Hilf (1971), Artificial paranoia, Artificial Intelligence, 2, 1-*25.

However, the notion that an understanding of human mental processes may necessarily be gained by studying the mechanisms within a computer program capable of simulating the behaviour of interest – and the potential for ethically dubious applications – provoked Weizenbaum, the designer of ELIZA, to a bitter reaction:

One of the really important socially relevant functions of computing is the explication of truly difficult concepts in the form of programming models. My own program ELIZA represented, according to some authorities, a major step toward the fulfillment of man's ancient dream of automating psychotherapy. The contribution here reported should lead to a full understanding of one of man's most troublesome disorders: infantile autism. Surely once we have a faithful and utterly reliable simulation of the behavioral aspects of this, or any other mental disorder, we understand it. How far away can the cure be then?

Joseph Weizenbaum (1974), Automating psychotherapy, Communications of the ACM, 17, 425.

Weizenbaum then proceeded to present a short program that responded to user inputs "*exactly* as does an autistic patient – that is, not at all".

46. Behaviourism: "objective natural science"

Psychology is concerned with understanding how what seem like perfectly natural mental operations, such as remembering, learning and reasoning, actually happen:

The object of psychology is to give us a totally different idea of the things we know best.

Paul Valéry (1943), Tel Quel.

Paul Valéry was a French poet who was influenced by symbolism, a movement not arguing for the merits of symbol-processing in AI but reacting against the dominant realist and naturalist tendencies in literature.

Psychology began as a would-be science in the nineteenth century when researchers asked people to perform various mental tasks and then to introspect, that is, look into their own thought processes, to explain how they

did them. For some tasks, such as answering the questions "Which letter of the alphabet comes before L?" and "What is the highest common divisor of 56 and 98?", the responses appeared to be reliable, but, in general, introspectionism failed because people often could not generate any satisfactory awareness, or even any awareness at all, of their own thought processes:

"He has a profound contempt for human nature."

"Of course, he is much given to introspection."

Charles-Maurice de Talleyrand (1754-1838).

Methodologically, obtaining post-hoc rationalisations seemed unsound, as thoughts were being used to examine thoughts.

So introspective psychology declined, although we can see some similarities with modern cognitive psychology:

The assumptions that underlie most contemporary work on information processing are surprisingly like those of nineteenth century introspective psychology, though without introspection itself.

Ulric Neisser (1976), Cognition and Reality: Principles and Implications of Cognitive Psychology, San Francisco: W.H. Freeman.

Instead of introspection, other techniques are used to access internal mental processes. For example, people are asked to talk during, not after, a problem solving process and what they say is then subjected to a detailed 'protocol analysis'. Of course, it is possible that talking interferes with the process being talked about. Other researchers look at external manifestations of the internal processes and try to interpret the former in terms of the latter. Examples are reaction time studies and the use of eye-tracking devices to obtain very precise records of what exactly on a screen an eye is directed towards.

The difficulties with introspection led to the conclusion that the idea that human behaviour could be best understood by speculating about internal mental processes should be dismissed as unscientific at a time when science was considered to be concerned only with the measurement of objective data:

Psychology as the behaviourist views it is a purely objective natural science. Its theoretical goal is the prediction and control of behaviour. Introspection forms no essential part of its method nor is the scientific value of its data dependent upon the readiness with which they lend themselves to interpretation in terms of consciousness. The behaviourist ... recognises no dividing line between man and brute.

John Watson (1913), Psychology as the behaviorist views it, Psychological Review, 20, 158-177.

One of the leading proponents of behaviourism, B.F. Skinner (1904-1990), was still arguing his case long after it had fallen out of favour:

> **We do not need to try to discover what personalities, states of mind, feelings, traits of character, plans, purposes, intentions or other perquisites of autonomous man really are in order to get on with a scientific analysis of behaviour.**
>
> *B.F. Skinner (1972), Beyond Freedom and Dignity, New York: Bantam.*

It should be emphasised that behaviourism was neutral to the question of whether or not such phenomena actually existed. The fact that they were mentioned at all suggested that they were assumed to exist. Behaviourists just argued that they should not be part of psychology.

As Skinner saw his life's work on behaviourism being put aside by the new paradigm of cognitive psychology, he resorted to sarcasm:

> **Cognitive psychology is frequently presented as a revolt against behaviorism, but it is not a revolt; it is a retreat. Everyday English is full of terms derived from ancient explanations of human behavior. We spoke that language when we were young. When we went out into the world and became psychologists, we learned to speak in other ways … But now … we can talk about love and will and ideas and memories and feelings and states of the mind, and no one will ask us what we mean; no one will raise an eyebrow.**
>
> *B.F. Skinner (1984), The shame of American education, American Psychologist, 39, 947-954.*

A similar reaction can be seen in the response of some symbol-processing cognitive scientists to reactive or behaviour-based theories, which they describe as a reversion to previously rejected forms of behaviourism. In fact, they might be considered a more extreme form, as the original was based on a methodological commitment to be concerned with only what can be observed, whereas behaviour-based AI argues that internal mental constructs not only cannot be observed but also do not exist at all.

In practice, however, behaviour-based robots appear to need internal representations such as maps, which is perhaps as well as it had been shown long ago in behaviourists' experiments that rats retain a memory of the layout of mazes. More recently, studies of the human brain have found that groups of neurons encode visual spaces. Consequently, despite the rhetoric, it seems that strict behaviourism is not tenable:

> **The strict computational behaviorist position for the modelling of intelligence does not scale to human-like problems and performance.**
>
> *John Tsotsos (1995), Behaviorist intelligence and the scaling problem, Artificial Intelligence, 75, 135-160.*

The vehemence of the arguments between the neo-behaviourists and the symbolic AIers may seem surprising until it is reflected that cognitive science and AI were themselves established only after fierce battles with the then-dominant behaviourists. The aversion that the founders of AI felt for the view that internal, mental constructs were irrelevant led naturally to an over-emphasis on the conviction that cognition occurred within individual minds, with a neglect of the external world, which behaviourists, by virtue of their concern with actual behaviour in the world, had to take into account.

Only recently has there been a return to some kind of balance, with the emphasis now rather less on a mind and a world but on a set of minds in the world, with interactions between them. Although most AIers regard behaviourism with disdain, it may be remarked that the Turing test is fundamentally behaviourist, being concerned only with external manifestations of presumed intelligent processes. The assumption or hope is that, as with many tests and examinations, it is not possible to pass the test by some 'fraudulent' process that somehow does not qualify as a measure of what is being assessed. Nonetheless, the test does emphasise the superficiality of *what* a system does, not the more fundamental question of *how* it does it.

47. Cognitive science: "one of the great revelations"

During the early years of AI many psychologists adopted AI's terminology, perhaps more as a convenience than a deep philosophical commitment. Discussions of human memory, for example, routinely drew analogies with computer memory, with concepts such as long-term memory being somewhat like backing store, with data transferred from it to working memory. The first to argue seriously that this analogy was more than a useful metaphor were Newell and Simon in their continuation of the GPS project:

> **The theory posits a set of processes or mechanisms that produce the behaviour of the thinking human ... Thus, the theory purports to explain behaviour – and not just to describe it ... Our theory posits internal mechanisms of great extent and complexity, and endeavours to make contact between them and visible evidences of problem solving. That is all there is to it.**
>
> *Allen Newell and Herbert Simon (1972), Human Problem Solving, Englewood Cliffs, N.J.: Prentice-Hall.*

Any new theory, for which the existing terminology is inadequate, is bound to use metaphor to enable itself to be understood but metaphor alone is no

basis for a science. So, to make progress, perhaps the metaphor needs to be taken literally:

> **Given that computation and cognition can be viewed in these common abstract terms, there is no reason why computation ought to be treated as merely a metaphor for cognition, as opposed to a hypothesis about the literal nature of cognition.**
>
> *Zenon Pylyshyn (1980), Computation and cognition: issues in the foundation of cognitive science, Behavioral and Brain Sciences, 3, 111-132.*

Metaphor is a rich area for linguistic study but surely nobody can decree that a metaphor become literal. Metaphors are described as being 'live' when they are used consciously as substitutes for literal equivalents, and 'dead' when they have been used so often that the original non-literal nature has been forgotten, as it has been, for example, with the word 'calculate', from the Latin word for a pebble used as a counter. Like us, metaphors have to work hard to change from being live to being dead. The 'pumping heart' is half-dead; the 'computing mind' has only recently been born.

Even though some psychologists felt the need to embrace AI, it does not follow that AIers have to concern themselves with psychology. In fact, though, the psychological importance of AI came to be a basic assumption of the field:

> **The fundamental working assumption or 'central dogma' of AI is this: What the brain does may be thought of at some level as a kind of computation.**
>
> *Eugene Charniak and Drew McDermott (1985), Introduction to Artificial Intelligence, Reading, Mass.: Addison-Wesley.*

It is worth pausing to reflect on the direction of this analogy. Most AIers, including Charniak and McDermott, are more computer scientists than brain scientists. If they had said that, as designers of computer programs, they would base their designs on the human brain, it being the only working (well, sufficiently) model for the desired computational behaviour, then that would have been a defensible research programme. But, no, the assertion is about the brain, even though, as their own textbook indicates, most AIers seem to know very little about the brain or at least seem to think that what is known is irrelevant to AI.

Still, those who do know more are happy to go along with the suggestion:

> **The mind is a system of organs of computation, designed by natural selection to solve the kinds of problems our ancestors faced in their foraging way of life, in particular, understanding and outmaneuvering objects, animals, plants, and other people … The discovery by cognitive**

science and artificial intelligence of the technical challenges overcome by our mundane mental activity is one of the great revelations of science.

Steven Pinker (1997), How the Mind Works, New York: W.W. Norton & Co.

Steven Pinker, of the Department of Brain and Cognitive Sciences at MIT, has swallowed the computational theory of mind – hook, line and thinker. The very title of his book begs the two questions that cognitive psychologists were most beginning to doubt: that the mind 'works', like a machine, and that the mind can be sensibly discussed in isolation, that is, separate from the body and the world. The book presents conventional cognitive psychology, combined with evolutionary biology, for the general reader. According to Pinker, AI has provided a 'great revelation' in helping to uncover the complexity of human mentality, through the technical challenge of duplicating it.

The name of cognitive science was given to this interdisciplinary marriage of (most of) AI and psychology, together with (some of) philosophy, linguistics and anthropology, focussed on the study of the mind and cognition, in order to make it clear that it was broader than psychology and moreover was a bona-fide science, which psychology had not quite managed to become. Although the name might have been new, the subject itself was not, as John Sowa remarked, echoing a familiar comment that all of philosophy is a footnote to Plato:

Aristotle was the founder of cognitive science: all the work on representing knowledge for the past 2300 years can be viewed as an application, refinement, extension, or re-invention of something that Aristotle either developed in detail or mentioned in passing.

John Sowa (1984), Conceptual Structures: Information Processing in Mind and Machine, Reading, Mass.: Addison-Wesley.

As Aristotle showed, it is possible to have a cognitive science without computers but the modern version was conceived as a twin of the computer and they shared a vigorous development.

Superficially, this seems in keeping with a long tradition of considering humans to be physical machines:

The idea that man is a machine is not a new one.

Piotr Ouspensky (1950), The Psychology of Man's Possible Evolution.

Piotr Ouspensky (1878-1947) founded The Society for the Study of Normal Man, which is an intriguing concept, and was a disciple of the enigmatic and influential Russian George Gurdjieff, whose diverse philosophy defies summarization in a phrase. But if a man is a machine, what sort of machine is he? Many of them, apparently:

There is a strong and, it seems, almost irresistible tendency in the human mind to interpret human functions in terms of the artifacts that take their place, and artifacts in terms of the replaced human functions. The power engine, with its levers and joints and its voracious fuel consumption, was a slaving giant, and, correspondingly, the human or animal body was a fuel-burning power machine. The modern servomechanism is described as perceptive, responsive, adaptive, purposive, retentive, learning, decision-making, intelligent, and sometimes even emotional (but this last only if something goes wrong), and, correspondingly, men and human societies are being conceived of and explained as feedback machines, communication systems, and computing machines. The use of an intentionally ambiguous and metaphorical terminology facilitates this transfer back and forth between the artifact and its maker.

Hans Jonas (1966), The Phenomenon of Life: Toward a Philosophical Biology, Chicago: University of Chicago Press.

Sometimes the analogy begins as just a convenient way of describing the new technology, as we saw with Babbage's and von Neumann's comments on their machines. Then sometimes the analogy becomes inverted: we begin to describe ourselves in terms of the new technology.

There are several possible kinds of reaction to the proposition that humans are just machines, and, in particular, computational machines:

Computers and men are not species of the same genus ... No other organism, and certainly no computer, can be made to confront genuine human problems in human terms.

Joseph Weizenbaum (1976), Computer Power and Human Reason, San Francisco: W.H. Freeman and Co.

The hypothesis that so-called simple physical laws govern every event in the body may be the correct theory, but it tells us nothing about the body as a system.

Heinz Zemanek (1974), The Human Being and the Automaton.

Of course the brain is a digital computer. Since everything is a digital computer, brains are too.

John Searle (1980), Minds, brains and programs, Behavioral and Brain Sciences, 3, 417-457.

Man is still the most extraordinary computer of all.

John F. Kennedy (May 21 1963).

In other words, we may flatly deny the proposition, or we may accept it but consider it useless or vacuous, or may accept it but still try to retain the superiority of humanity.

Or we may consider the proposition to be a profound theoretical contribution. Cognitive science is concerned with internal mental processes and with the hypothesis that these are computational, in some sense. The main manifestation has been the symbol-processing approach, with the contribution of connectionism ebbing and flowing. The association of cognitive science with symbolic AI, that is, with the assumption that computational models such as production systems, schemata, semantic networks, and the like, map onto an internal mental level of processing, became almost unquestioned:

> **The central hypothesis of cognitive science [is that] thinking can best be understood in terms of representational structures in the mind and computational procedures that operate on those structures.**
>
> *Paul Thagard (1997), Mind: Introduction to Cognitive Science, Cambridge, Mass.: MIT Press.*
>
> **The central themes that emerged … and that mark the cognitive sciences are the concepts of representation and process. They are the primary foci of all the relevant disciplines, and it is symptomatic of our acceptance and their importance that we rarely hear anybody question these two foundations.**
>
> *George Mandler (1984), Cohabitation in the cognitive sciences, in Walter Kintsch, James Miller and Peter Polson (eds.), Methods and Tactics in Cognitive Science, Hillsdale N.J.: Erlbaum.*

Today, these foundations of representation and process certainly are questioned, although it remains the view of some that computational psychology remains the best bet for explaining cognition:

> **[It] is, in my view, by far the best theory of cognition that we've got ... [Its] central idea – that intentional processes are syntactic operations defined on mental representations – is strikingly elegant. There is, in short, every reason to suppose that Computational Theory is part of the truth about cognition.**
>
> *Jerry Fodor (2000), The Mind Doesn't Work That Way: The Scope and Limits of Computational Psychology, Cambridge, Mass.: MIT Press.*

However, this praise turns out to be faint indeed, for Fodor goes on to argue that computational psychology is inadequate for the most interesting and important aspects of the human mind. According to Fodor, it may have some explanatory power in the areas of perception and language but the frame problem (mentioned earlier and which he defines as the making of reliable abductive inferences efficiently) renders the theory useless for the processes of higher cognition.

Undaunted by growing criticisms of the symbol-processing basis of cognitive science, one of its founders felt sufficiently bold to suggest that enough had been achieved that it was time to unify all the results into a single theory of cognition, capable of explaining all of human cognition:

Psychology has arrived at the possibility of unified theories of cognition – theories that gain their power by positing a single system of mechanisms that operate together to produce the full range of human cognition.

Allen Newell (1990), Unified Theories of Cognition, Cambridge, Mass.: Harvard University Press.

As an illustration of such a unified theory, Newell quoted his own Soar, naturally, although the ACT* system developed earlier by John Anderson, his Carnegie-Mellon colleague (or perhaps not, as they seem never to have co-published) had also been intended to be a unitary theory of mind:

In ACT* the same core system if given one set of experiences develops a linguistic facility, if given another set of experiences develops a geometry facility, and if give another set of experiences develops a programming facility. Therefore ACT* is very much a unitary theory of mind.

John Anderson (1983), The Architecture of Cognition, Cambridge, Mass.: Harvard University Press.

48. Neural models: "a new embodied view"

Consistent with his view that a unified theory of cognition needs to be expressed at the symbol or knowledge level, Newell paid very little regard to neuroscience. Others, of course, consider the physical brain and how it functions to be the essence of cognition. Those theorists who were seeking functional equivalences at much lower levels, at almost the hardware level, were not so interested in representations. They compared neural mechanisms with processes occurring within computers:

Programs of machines are of a linear character. That is, at any given moment impulses move sequentially from one element to the next ... Living organisms function in a completely different way. In these systems a huge number, literally millions of programs, are realized simultaneously ... Methods are now being developed for parallel information processing according to several programs … Without doubt, in the near future well-developed informational systems will be created, systems which will be capable of exceeding the limits of the human brain.

Nikolai Amosov (1965), Modelling of Thinking and the Mind, New York: Spartan Books.

The technological development of parallel processors has weakened the criticism that cognition cannot be computational because computers are necessarily serial and human cognition is considered to be, at least in part, parallel.

The advent of parallel systems is considered by some to provide an escape from the von Neumann model of computation that has blinkered our thinking on the nature of models of intelligence:

> **The emergence of massively parallel computers liberates the way intelligence may be modelled … massively parallel artificial intelligence will be the central pillar in the ultimate success of artificial intelligence in both engineering and scientific goals.**
>
> *Hiroaki Kitano (1993), Challenges of massive parallelism, Proceedings of the 13th International Joint Conference on Artificial Intelligence, 813-834.*

To emphasise the magnitude of the technological advance, these systems are invariably described as 'massively parallel', never ordinarily so. Engineering benefits will doubtless ensue and there is certainly the potential for massively parallel computation to provide better models of some aspects of human cognition, for example, of the speed of human face recognition, but so far massive parallelism has not provided a theoretical breakthrough for cognitive science.

A rather more fundamental objection to the use of information processing models for the brain is made by Gerald Edelman, who uses the old philosophical argument of the 'homunculus fallacy', which is that the method involves ascribing to internal devices the very psychological properties which are being investigated. According to Edelman, it is acceptable to use such models as a post-hoc explanation or description of the brain but they cannot be ascribed to the brain itself:

> **The selective behavior of ensembles or neuronal groups may be describable by certain mathematical functions … but it seems as unlikely that a collection of neurons carries out the computation of an algorithm as that interacting lions and antelopes compute Lotka-Volterra equations.**
>
> *Gerald Edelman (1987), Neural Darwinism: The Theory of Neuronal Group Selection, New York: Basic Books.*

The Lotka-Volterra equations are used to describe behaviour in predator-victim situations. Edelman's point is that however accurately these equations describe the actual behaviour observed it is absurd to imagine that the participants compute such equations to determine their behaviour.

That, at least, is the reaction one is intended to have to the absurd thought of lions and antelopes computing the fearsome-sounding Lotka-Volterra equations. In fact, the equations merely state that if V and P are the

densities of victim and predator, then the rate of change of victims is determined by the birth rate of victims and the kill rate (or V' = bV – kPV) and the rate of change of predators by the 'conversion rate' of kills to predator births and the death rate of predators (or P' = ckPV – dP). So, the Lotka-Volterra equations are not concerned with the *behaviour* of predators or victims at all. They provide only a *summary* of the outcome of such behaviour. They are therefore not comparable to the processes studied in cognitive science. It is indeed absurd to imagine lions and antelopes (especially antelopes) wasting time computing such equations.

Edelman won the 1972 Nobel Prize for medicine for showing that the immune system works in a Darwinian way. His theory of the brain is apparently similar: neuronal groups, that is, randomly connected brain cells, grow and evolve in response to external stimuli. The details are complicated – too complicated for eminent neuroscientists like Francis Crick and Gunther Stent, it seems, so AIers have little chance of understanding the theory. From his theoretical high ground, Edelman delights in provoking AIers by insisting that he has generated a crisis for computational cognitive science, which, in his view, is wrong-headed. Most AIers see the theory as arguing for a different kind of computation – more analogue, dynamic and non-deterministic – rather than for no computation at all.

To muddy matters even further for simple AIers, Edelman proceeds to demonstrate his theory by means of computer simulations. Well, a computer simulation of, say, a telephone network would not show that that network was a computer, but if neural Darwinism can be demonstrated by a computer simulation then that would be some evidence that the mind *could* be computational.

At least, it is very possible that the symbol-processing methods of standard AI provide a very sanitised basis for any theory of human cognition. AI has contrasted logical, rational reasoning with emotion and intuition, without acknowledging that the latter underpins and motivates the former. It has tended to isolate the individual mind from the social context that provides the rationale for the mind to function. In short, it may have eliminated from consideration that which matters most:

> **Our cognitive sciences are themselves suffering from an agnosia essentially similar to Dr P's. Dr P may therefore serve as a warning and parable – of what happens to a science which eschews the judgemental, the particular, the personal, and becomes entirely abstract and computational.**
>
> *Oliver Sacks (1987), The Man Who Mistook His Wife for a Hat, New York: Harper & Row.*

Oliver Sacks, the neurologist and author, warned AIers of the dangers of an overly narrow focus in his account of the man (Dr P) who mistook his wife for a hat. Dr P recognised objects by a feature matching process reminiscent of AI programs.

Cognitive science developed with the assumption that the mind of the individual was the appropriate unit of analysis, just as computer science developed with the assumption that the Turing machine, that is, an isolated computer with a single program, was the theoretical base from which all computer analysis could be derived. Recent work may be indicating that both assumptions were unsound and have led to an undue emphasis on some aspects of cognition and computation and a neglect of more important ones. If cognitive science had begun with 'group cognitions' as fundamental, on the reasonable basis that humans are first and foremost social beings, with individual cognitions emerging as a special, limited case, then it would be a very different subject and many of the controversies, such as those surrounding situated cognition, would not have arisen, at least, not in the confrontational form that they did. If computer science had begun with the idea that computers are essentially devices for communicating with other (human and artificial) computers – which they are – then AI might not have developed with its emphasis on self-contained symbol-processing, which is now a paradigm from which researchers are attempting to escape.

Peter Wegner, a long-time commentator on the design of programming systems, has argued that as Turing machines cannot handle the passage of time or interactive events that occur during the process of computation they are not as powerful computing mechanisms as concurrent and non-terminating reactive processes, as theoreticians have known since the 1970s. Therefore, he argues, the Turing machine model should be replaced by an 'interaction model':

> **The insight that interactive models of empirical computer science have observably richer behavior than algorithms challenges accepted beliefs concerning the algorithmic nature of computing, allowing us to escape from the Turing tarpit and to develop a unifying interactive framework for models of software engineering, AI, and computer architecture.**
>
> *Peter Wegner (1997), Why interaction is more powerful than algorithms, Communications of the ACM, 40, 5, 80-91.*

Unfortunately but probably inevitably, these interactive frameworks do not have the simplicity of Turing machines.

An indication of the possibly transitional state of cognitive science can be gained from a spat resulting from the publication of *The MIT Encyclopedia*

of the Cognitive Sciences, a 1312-page reference work on cognitive science published in 1999. According to George Lakoff:

> **The formalist nativist paradigm with which cognitive science began in the 1960s and early 1970s has been turned on its head. In place of logic, there are image-schemas, frames, metaphorical mappings, mental spaces, and so on … In place of symbol systems, there are highly structured neural models. In place of Anglo-American analytic philosophy with its correspondence theory of truth, there is emerging a new embodied view of philosophy with an embodied account of truth … Unfortunately, you can read virtually nothing about all these exciting developments in cognitive science from reading *The MIT Encyclopedia of the Cognitive Sciences.***
>
> *George Lakoff (2001), As advertised: a review of The MIT Encyclopedia of the Cognitive Sciences, Artificial Intelligence, 130, 195-209.*

According to this view, recent research on embodied cognition has overthrown the disembodied symbol-processing approach attributed to Newell, Simon, McCarthy and others and has contradicted the idea that mind can be studied independently of the physical body, assumed to be a tenet of AI. Naturally, the Encyclopedia editors, supported by other more generous reviews, consider Lakoff's views "antithetical to the ecumenicism" of modern cognitive science. Although not clear what "formalist nativist" means, they are sure that there is no bias towards it. Lakoff's attempt to subvert established cognitive science is dismissed as "a yawningly dated view of where the cognitive sciences are at". Instead, it is argued that results on embodied cognition are more modest than claimed and anyway have been incorporated into the broad church of contemporary cognitive science. As always, we shall see how this pans out.

49. Analytical philosophy: "uniformly negative"

Just as some psychologists considered that AI provided a revolutionary new tool for the field of psychology, so some philosophers turned to AI in the hope that it would help clarify concepts that they had discussed inconclusively heretofore:

> **Philosophers have been struggling for centuries to develop techniques for articulating commonsense and unacknowledged presuppositions, such as the techniques of conceptual analysis and the exploration of paradoxes. AI provides an important new tool for doing this. It helps us find our mistakes quickly ... 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.

Aaron Sloman (1978), The Computer Revolution in Philosophy: Philosophy, Science and Models of Mind, Brighton: Harvester Press.

Indeed, the revelatory process of developing a computer model of a theory of some mental activity, with the inevitable discovery that the theory is inadequate, leading to subsequent revision of the theory (as opposed, presumably, to armchair philosophy) has led some to believe that computer modelling will be indispensable to future epistemologists, that is, those engaged in the theory of knowledge:

Philosophers may not fully appreciate what a powerful technique this [computer modelling] is until they try it, but once they try it they will never be content to do without it. I predict that this will become an indispensable tool for epistemology over the next twenty years.

John Pollock (1989), How to Build a Person: A Prolegomenon, Cambridge, Mass.: MIT Press.

John Pollock, Professor of Philosophy at the University of Arizona, has certainly taken himself at his word, embarking on an odyssey to create an artificial person, anticipating philosophical vistas along the way.

How, exactly, does AI help philosophers?:

What makes AI an improvement on earlier philosophers' efforts at model sketching … is the manner in which skepticism is vindicated: by the actual failure of the system in question.

Daniel Dennett (1988), When philosophers encounter artificial intelligence, in Stephen Graubard (ed.), The Artificial Intelligence Debate, Cambridge, Mass.: MIT Press.

So, the role of AI in philosophy is, in the Popperian style, to provide evidence for the inadequacies of proposed theories, through the mistakes and failures exposed. Certainly there are plenty of these, but it isn't always the case that they have philosophical significance. The belief appears to be that AI provides a more efficient means to philosophical enlightenment than the earlier methods of thinking, reading, discussing and writing:

The optimistic view, of course, is that artificial intelligence researchers can make much faster progress than all those philosophers because we are equipped with 'powerful ideas' they didn't have, especially the idea of sophisticated autonomous computation. I hope this is right. But if all we do is go on writing programs, without any general theories emerging, then I am going to get increasingly uncomfortable.

Drew McDermott (1987), A critique of pure reason, Computational Intelligence, 3, 151-237.

Drew McDermott must now be comfortable in the sense that hospital patients who have just survived a serious operation are said to be comfortable. McDermott began in AI working on Planner-like languages and developed his reputation through research on non-monotonic logics and planning. His forthright views on AI had already been indicated by his 1976 paper *Artificial intelligence meets natural stupidity*, in which he berated wishful-thinking AI, when in his *A critique of pure reason* of 1987 he recanted his own previous beliefs on the role of logic in AI. Now director of the Yale Center for Computational Vision and Control, he advocates dispensing with complex knowledge representations in favour of tractable special-purpose techniques (as we have seen, for example, in planning, learning, and speech-recognition), which will be adequate for all the real-life problems that actually need addressing.

Perhaps pre-eminent among "all those philosophers" to whom McDermott refers is René Descartes (1596-1650), whose ideas laid the foundations for modern scientific investigation and of mathematical physics, analytic geometry, and the philosophy of mind. Today, his famous formula expressing a sufficient, if not necessary, condition for existence is not universally respected:

> ***I think, therefore I am* is the statement of an intellectual who underrates toothaches.**
>
> *Milan Kundera (1991), Immortality.*

In his lifetime, Descartes' ideas were dangerously revolutionary, because the church taught beliefs as undisputable truths. Partly as a result, he moved from France to Holland, where the church had less influence, and in 1649 to Sweden, where Queen Christina had invited him to teach her philosophy (imagine a queen needing philosophy today!). Unfortunately, the delicate health of Descartes proved no match for the Swedish winter, which perhaps illustrated perhaps his philosophy of dualism, according to which there were two fundamentally different modes of existence: 'thinking' things (minds) and 'extended substances' (matter).

Descartes was the father of analytic philosophy, which is the dominant tradition in Western philosophy and with which standard AI is in tune. Analytic philosophy is based upon logical and linguistic analyses, which have attempted to synthesise rationalism (the theory that we can derive knowledge of the world by pure reason) and empiricism (the theory that experience is a necessary basis for all knowledge). Analytic philosophy is sometimes equated with logical positivism, although the latter term is best restricted to refer to a logical or scientific form of empiricism.

Descartes, like Plato and many other philosophers, was a great mathematician and while it would be nice to imagine that philosophical ideas arise, and should be judged, independently of context, it is surely no coincidence that both regarded mathematics as the basis of all human knowledge, when, in fact, the definitional nature of truth in mathematics is quite unlike that in other forms of knowledge, such as moral, aesthetic, religious, and even scientific knowledge. In particular, the search for certainty, which was inaugurated by Descartes, is now considered to be mistaken, since good theories no longer aim to be true but only better than alternatives.

Anyway, Descartes began the rationalist side of analytic philosophy, with Locke, Berkeley and Hume on the empiricist side, the two sides being synthesised later by Kant and others. If there is to be analysis, it needs to be agreed what the analysis is to be of. The two obvious contenders are mind and language, echoing the behaviourist's distinction between internal and external or observable phenomena. Views about the primacy of mind and language have oscillated within the analytic tradition.

Initially the function of language was considered to be simply to enable the transfer of ideas from one mind to another. Then analytic philosophers came to consider language as primary, with mental events being seen as just dispositions to verbal behaviour. Today, there is a reversion to the earlier view that mind is more fundamental than language:

Thoughts die the moment they are embodied by words.

Arthur Schopenhauer.

I insist that words are totally absent from my mind when I really think.

Jacques Hadamard (1945), The Psychology of Invention in the Mathematical Field, Princeton, N.J.: Princeton University Press.

The words or the language as they are written or spoken do not seem to play any role in my mechanism of thought.

Albert Einstein (1954), Ideas and Opinions, London: Souvenir Press.

We are getting into semantics again. If we use words, there is a very grave danger they will be misinterpreted.

H.R. Haldeman (1973), testifying in his own defence, Watergate trial.

Of course, the interrelationship between thought and language is complex. For example, studies show that the language we use influences our thought processes, indicating that the quotations above, suggesting that words are irrelevant to, or even interfere with, thought, are too simplistic.

During the period when the emphasis was on linguistic analysis, it came to be believed that formal languages, such as mathematics and logics, were the best means by which philosophical knowledge might be gained:

> **Modern analytical empiricism ... differs from that of Locke, Berkeley and Hume by its incorporation of mathematics and its development of a powerful logical technique ... I have no doubt that, in so far as philosophical knowledge is possible, it is by such methods that it must be sought.**
>
> *Bertrand Russell (1961), History of Western Philosophy, London: Allen and Unwin.*

Philosophical knowledge is, of course, a form of knowledge and Russell's assertion that it should be expressed in mathematical and logical form is a continuation of the traditional view of knowledge and reason since the Greeks.

As far as AI is concerned, it came to be the central assumption of the field that an intelligent artifact had to 'possess knowledge' about its world and had to reason to infer further knowledge:

> **If artificial intelligence researchers can agree on anything, it is that an intelligent artifact must be capable of reasoning about the world it inhabits. The artifact must possess various forms of knowledge and beliefs about its world, and must use this information to infer further information about that world in order to make decisions, plan and carry out actions, respond to other agents, etc.**
>
> *Raymond Reiter (1987), Nonmonotonic reasoning, Annual Review of Computer Science, 2, 147-186.*

The only thing AIers agree on is that there is nothing they can agree on. As we have seen, since Reiter wrote this in 1987 there have been increasing denials that intelligent artifacts need to reason and that such artifacts need to possess knowledge. They can just react to their environment, it is claimed. Nonetheless, the knowledge-reasoning paradigm remains the prevailing view in AI:

> **If we want to design an entity … capable of behaving intelligently in some environment, then we need to supply this entity with sufficient knowledge about this environment. To do that, we need an unambiguous language capable of expressing this knowledge, together with some precise and well understood way of manipulating sets of sentences of the language which will allow us to draw inferences, answer queries, and update both the knowledge base and the desired program behavior.**
>
> *Michael Gelfond and Nicola Leone (2002), Logic programming and knowledge representation: the A-Prolog perspective, Artificial Intelligence, 138, 3-38.*

– a statement which, offered fifteen years later, is a virtual paraphrase of Reiter's opinion.

Unfortunately for AI, its enthusiastic adoption of the analytic tradition occurred at just the time analytic philosophy was being increasingly questioned. Hubert Dreyfus, a trenchant critic of AI, whose regular re-

cycling since 1965 of his *What Computers (Still) Can't Do* polemic indicates that AI has remained sufficiently alive to warrant a further battering, conceded that AI represented a culmination of the dominant, analytic tradition, but, anticipating the failure of AI, argued that if AI should turn out to be impossible then this would show that analytic philosophy was unsound:

> **Aristotle defined man as a rational animal, and since then reason has been held to be of the essence of man. If we are on the threshold of creating artificial intelligence we are about to see the triumph of a very special conception of reason. Indeed, if reason can be programmed into a computer, this will confirm an understanding of the nature of man which Western thinkers have been groping toward for two thousand years but which they only now have the tools to express and implement. The incarnation of this intuition will drastically change our understanding of ourselves. If, on the other hand, artificial intelligence should turn out to be impossible, then we will have to distinguish human from artificial reason, and this too will radically change our view of ourselves.**
>
> *Hubert Dreyfus (1972), What Computers Can't Do: a Critique of Artificial Reason, New York: Harper and Row.*

Dreyfus knew that he was on safe ground in binding AI and analytic philosophy, so that the failure of one would imply the failure of the other, for philosophers themselves were already doubting the thesis of analytic philosophy, and indeed, in a broader context, these doubts can be seen as paralleling the destabilising influence of ideas such as the uncertainty principle in physics and undecidability in mathematics and logic:

> **Not a single one of the great positive theses of Logical Empiricism ... has turned out to be correct. It detracts from the excitement of the fact that, by turning philosophical theses into linguistic ones one can make philosophy more scientific and settle the truth value of philosophical propositions by hard scientific research, if the results one obtains are uniformly negative.**
>
> *Hilary Putnam (1975), Mind, Language and Reality: Philosophical Papers Vol. 2, Cambridge: Cambridge University Press.*

50. Epistemology: "like woof and warp"

Analytic philosophy is not directly concerned with epistemology, that is, the nature of knowledge, which might be considered to be the central concern of AI systems. It is a methodology that suggests that the way to obtain knowledge is through formal analysis – it does not follow that the knowledge itself is of a formal, analytic nature. However, it is easy to conflate the

formulae written for us to analyse as philosophers and the object of those formulae, the knowledge used by human reasoners in general. George Lakoff characterised analytic philosophy's view of knowledge and reasoning as follows:

> **Modern attempts to make it work assume that rational thought consists of the manipulation of abstract symbols and that these symbols get their meaning via a correspondence with the world, objectively construed, that is, independent of the understanding of any organism ... A collection of symbols placed in correspondence with an objectively structured world is viewed as a *representation* of reality ... Thought is the mechanical manipulation of abstract symbols. The mind is an abstract machine, manipulating symbols essentially in the way a computer does, that is, by algorithmic computation.**
>
> *George Lakoff (1987), Women, Fire, and Dangerous Things: What Categories Reveal about the Mind, Chicago: Chicago University Press.*

Lakoff is here summarising the view of others, a view he then sets out to show is mistaken. The philosophy, which Lakoff calls objectivism, that there is a reality 'out there' in the world, that is, external to and independent of ourselves, which we must seek to represent, is deeply embedded in Western thought – so deeply, in fact, that it is difficult to put it aside and think of alternatives or to think in alternatives.

But even in analytic philosophy it may be accepted that the scientific search for a single, presumably correct, representation of reality is misguided. It is possible for individuals to interpret the world differently to come up with valid representations that are not alike. It is even possible for an individual to hold conflicting views:

> **I have opinions of my own – strong opinions – but I don't always agree with them.**
>
> *George W. Bush (2002).*

If we allow the possibility of different representations of the world then we must allow the same of mental worlds, and hence accept that different philosophies of knowledge are possible, without assuming that only one of them is necessarily 'correct'.

We could deny that knowledge or meaning exists objectively in the world, to be discovered by us. Instead, we could consider that individuals construct meaning in their attempts to make sense of the world. This meaning depends not only on the situation they are in but also on their purposes and activities and on their prior conceptions of the world. The constructions are tentative models that are continually tested against experience and modified, if necessary:

> **Constructivism, founded on Kantian beliefs, claims that reality is constructed by the knower based upon mental activity. Humans are perceivers and interpreters who construct their own reality through engaging in those mental activities … What the mind produces are mental models that explain to the knower what he or she has perceived … We all conceive of the external reality somewhat differently, based on our unique set of experiences with the world and our beliefs about them.**
>
> *David Jonassen (1991), Objectivism vs constructivism: Do we need a new philosophical paradigm? Educational Technology, Research and Development, 39, 3, 5-13.*

Constructivism is therefore a philosophy of learning and only indirectly one of knowledge. It exists in various flavours. If the tentative constructions are considered to be symbolic and retained in the human memory then the only difference from objectivism lies in the status of the symbolic representations: now they are considered to be temporary and adequate for an individual's purposes, and not intended to be objectively correct. Another distinction concerns whether knowledge or meaning is essentially an individual or a group concept. Social constructivists emphasise that knowledge emerges from interactions or dialogue within a group and is not necessarily possessed by any individual member. A further debate concerns the nature of this possession of knowledge, or the location of the constructed models. Situationists argue that, since knowledge is created dynamically in response to the enveloping situation, knowledge cannot be considered to reside in human memory, represented by symbolic structures.

The notion that knowledge doesn't exist in memory but in the situation may seem counter-intuitive, but consider the following scenario: Somebody is performing a task (say, editing a file), rather badly. You know how to do it better and are giving advice, from a distance. You find it difficult to recall how to do the job, but when you take over the task all your skills are immediately available. Somehow, the environment itself is necessary for triggering and making useful the knowledge that you have.

This kind of experience, which has, of course, been backed up by studies of how people appear to be able to perform tasks in certain environments and not in others has led to the view that knowledge is not a thing but a capability. People are able to perform tasks, for which they may or may not have been trained, but not be able to answer questions or provide explanations about the task. Knowledge seems then to arise from the interaction between internal constructs and the external environment:

> **Knowledge is fundamentally a production of the mind and the world, which like woof and warp need each other to produce texture and to**

> **complete an otherwise incoherent pattern. It is impossible to capture the densely interwoven nature of conceptual knowledge in explicit, abstract accounts.**
>
> *John Seely Brown, Allan Collins, and Paul Duguid (1988), Situated cognition and the culture of learning, Educational Researcher, 18, 1, 32-42.*

In this view, knowledge cannot be explained exclusively in terms of either internal constructs or the external environment. Some tasks (such as solving algebra problems) seem to require more of the former but some (such as fly fishing) more of the latter.

Some theorists emphasise the internal constructs, as the traditional AI endeavour has; some emphasise the environment:

> **[There is a] potentially radical shift from invariant structures to ones that are less rigid and more deeply adaptive. One way of phrasing this is to say that structure is more the variable outcome of action than its invariant precondition ... It (i.e. behaviour) involves a prereflective grasp of complex situations, which might be reported as a propositional disposition, but is not one itself … Learning is a process that takes place in a participation framework, not in an individual mind.**
>
> *W.F. Hanks (1991), Preface to Jean Lave and Etienne Wenger, Situated Learning: Legitimate Peripheral Participation, New York: Cambridge University Press.*
>
> **Rather than thinking that knowledge is in the minds of individuals, we could alternatively think of knowledge as the potential for situated activity. On this view, knowledge would be understood as a relation between an individual and a social or physical situation, rather than as a property of an individual.**
>
> *James Greeno (1989), Situations, mental models and generative knowledge, in David Klahr and Kenneth Kotovsky (eds.), Complex Information Processing: The Impact of Herbert A. Simon, Hillsdale, N.J.: Erlbaum.*

As these and other comments make clear, the idea that knowledge and learning are not properties of individuals is put forward as a radical alternative, intended to overthrow the tradition, traceable to Aristotle and Descartes, that the mind (which is where cognition was supposed to happen) must be separated from the world.

It is necessary to keep a sense of proportion in these musings on the nature of knowledge. The view that all knowledge is conjectural and socially constructed does not correspond to decision-making in the real world. Consider the following examples. Astronomers predict when lunar eclipses will occur for centuries ahead and so far these predictions have been accurate to the second. Of course, it is possible to imagine circumstances that will invalidate those predictions but overall the evidence indicates that there is a

world out there that can become known. We also know that human beings have two kidneys. Again, we can imagine exceptions (somebody may have had a kidney removed, or there may be a freak of nature). But a surgeon who mistakenly removes a healthy kidney instead of the diseased one, as in a recent case, would not be able to defend herself on the philosophical grounds of 'how was I supposed to know he only had two kidneys?'.

Every day we entrust our lives to beliefs that others know things. At traffic lights we rely on our perceptions and a common knowledge of the rules to interpret them. We do not speculate overmuch on whether others see the same colours as we do or creep over the crossing in case other drivers know different rules. Pathological cases will occur and need to be adapted to but intelligent agents (human and artificial) generally have to make do with a commonsense view of the nature of knowledge.

What, then, is the philosophical status of the 'knowledge' in knowledge-based systems? In retrospect it is now clear that a lot of needless confusion was caused by the keenness of knowledge engineers to go one up on database engineers and to establish so-called knowledge bases, in which knowledge was supposed to be inventoried. For a start, who can fail to be impressed by a knowledge base that has 10,000 rules, compared to one with a measly 1,000 rules?

But what exactly is a 'rule'? A rule such as "tingling sensation in the hands implies cervical spondylosis", or however it is expressed in program code, is not in itself knowledge, no more than is a formula such as "F = ma" or "$n^2a^2 = k^2\mu$". Almost everybody knows of Newton's third law of motion and may have come to think that "F = ma" is indeed what they know but for anyone unfamiliar with Kepler's third law it will be clear that the second formula cannot itself be knowledge. At best, it is a 'representation of knowledge' – or at least a representation of something. It is a description of something which may be of use in a particular situation but is not itself a capability.

It is natural for AIers to focus on the representation of internal constructs (since that is what programming is about) and to neglect the role of the environment, but it is a category mistake to consider the 'representation of knowledge' to be 'knowledge':

The map is not the territory.

Alfred Korzybski (1933), Science and Sanity: An Introduction to non-Aristotelian Systems and General Semantics.

Korzybski's phrase was intended to indicate that we should not confuse the map of reality that we carry around in our heads with reality itself. Although some philosophers would deny the 'reality' and that we carry such a map in

our heads, Korzybski, who fits the profile of the exotic, inscrutable and deceased European that is beloved of AIers, would seem ripe for re-discovery. He was a Polish count who in 1938 set up an Institute for General Semantics to develop tools that enable us to develop awareness of our own map-making process and hence make more appropriate responses to events around us.

51. The mind-body problem: "as digestion is to the stomach"

The situationist's view is a rebuttal of Descartes' fundamental philosophical contribution, his theory of dualism. In general, dualism is any doctrine that divides everything into two categories. In Descartes' case, the two categories are matter, consisting of physical bodies that have shape and size and may move, and minds, which have beliefs, hopes, emotions, and the like:

> **.. as I grew up I became increasingly interested in philosophy, of which they profoundly disapproved. Every time the subject came up they repeated with unfailing regularity, 'What is mind? No matter. What is matter? Never mind.' After some fifty or sixty repetitions, this remark ceased to amuse me.**
>
> *Bertrand Russell (1956), Portraits from Memory and Other Essays, London: George Allen & Unwin.*
>
> **Mind or matter, never mind, it does not matter.**
>
> *Nelson Goodman (1984), Of Mind and Other Matters, Cambridge, Mass.: Harvard University Press.*

Common sense would go along with the mind-matter distinction: physical bodies, like chairs and stones, have shape and size and don't have beliefs and emotions; beliefs and emotions don't have spatial properties. However, if they are completely separate, how can they be associated, as they seem to be when a desire leads to an action or a movement leads to a belief? More to our specific point, if we agree with dualism that mind and matter are two entirely different kinds of thing, how can a particular piece of matter, namely, a computer, be said to possess a mind in the sense that it entertains thoughts, beliefs and (possibly) emotions?

If one of AI's aims is to study the extent to which mind may be divorced from matter, in that intelligence may be shown by entities regardless of material form, then AI would seem to be fertile ground for a philosophical investigation of the mind-body problem. In fact, some commentators, overwhelmed by the success of the new cognitive science, considered that the mind-body problem had already been solved:

> **The mind-body problem ... is one that information-processing theory does answer. What has seemed to philosophers to be mind – a different sort of stuff from the brain – is not a separate stuff at all, but a series of processes of immense complexity, the integration of millions or billions of neural events ... A computer has no soul but only tangible parts, yet by means of its programs, it can simulate certain aspects of human thought. So, too, with our mind: it is not something apart from the brain, but is the brain's programs ... mind is to the brain as digestion is to the stomach.**
>
> *Morton Hunt (1982), The Universe Within: A New Science Explores the Human Mind, New York: Simon and Schuster.*

While we digest this analogy, we should acknowledge that this is not just a conclusion of cognitive science enthusiasts: for example, a leading biologist specialising in the functioning of the brain had already come to a similar conclusion:

> **The lives of human beings and other animals are governed by sets of programs written in their genes and brains ... The brain operates in certain organized ways that may be described as programs, and the actions of these programs constitute the entity that we call the mind of a person.**
>
> *J.Z. Young (1978), Programs of the Brain, Oxford: Oxford University Press.*

Young emphasised that he was using the word 'program' in its original sense of 'a plan decided beforehand to achieve some end' and that 'brain programs' are only partly like computer programs.

Indeed, he was anxious to adopt the concept of a program and yet dissociate himself from the negative connotations of computer programming, which he saw only in the narrow, conventional sense as a rigid, logical, deterministic activity:

> **Logic, mathematics, and computers represent potentially dangerous reductionisms.**
>
> *J.Z. Young (1978), Programs of the Brain, Oxford: Oxford University Press.*

Reductionism, often used as a term of abuse, is the view that any complex system, such as a computer program or the human brain, can be completely understood and explained in terms of its simpler parts, such as electronic components or neurons, respectively. We know that the behaviour of a computer is entirely determined by its electronic components and its programs, because we have built it that way. However, in practice, the behaviour of no computer program, certainly no AI program, is ever explained in terms of electronics and it is doubtful that it is even possible in principle.

If it seems surprising, in view of the amazing nature of computer technology, that its electronics are theoretically irrelevant, we should recall

the properties of the Turing machine, which make no reference to the means of implementation. Similarly, any analogy that is drawn between the computer and the brain is not necessarily implying that brain behaviour can be completely explained in terms of neural activities.

The view that AI and cognitive science have solved the mind-body problem for philosophers is now almost a commonplace:

> **The mind is what the brain does; specifically, the brain processes information, and thinking is a kind of computation … [The] computational theory of mind … is one of the great ideas of intellectual history, for it solves one of the puzzles that make up the "mind-body problem": how to connect the ethereal world of meaning and intention, the stuff of our mental lives, with a physical hunk of matter like the brain.**
>
> *Steven Pinker (1997), How the Mind Works, New York: W.W. Norton & Co.*

However, like Young, Pinker clarified that the computational theory of mind did not refer to real computers or real programs. This is something of a mystery, for they do not specify what kind of imaginary computer or program they have in mind for mind, nor in what theoretical respects they differ from real ones. If computational means computable by a universal Turing machine, is it argued that the mind's programs can all be computed, in principle, by real computers, which are theoretically equivalent to Turing machines? – or not?

Perhaps they have in mind the new connectionist machines rather than the older symbol-processing ones, for they appear to offer some promise:

> **It is likely that [connectionism] will offer the most significant progress of the past several millennia on the mind-body problem.**
>
> *Paul Smolensky (1988), On the proper treatment of connectionism, Behavioral and Brain Sciences, 11, 1-23.*

This progress remains unclear, not least because connectionist machines can be simulated on Turing machines, so their theoretical basis for any improvement is questionable.

The position of the adjective in 'the computational theory of mind' needs to be noted: it is the theory that is said to be computational, not the mind. If we had a 'mathematical theory of mind' we would expect the theory to be expressed in a mathematical formalism, with no implication that the mind did maths. So, a computational theory of mind should be expressed in a computational formalism, with no implication that the mind did computations. But it is not: the theory is described informally in English with reference to postulated computational processes in the mind.

52. Determinism: "we have no choice"

If AI might be considered to have dispensed with the mind-body problem, or at least to have shed fresh light on it, perhaps it could do the same with other key philosophical issues, such as that of determinism, that is, the thesis that any event is an instance of some law of nature and therefore predicted by it, so that the unfolding of events is determined by those laws of nature, and that, by extension, human beings are themselves subject to deterministic processes and hence have no free will:

An intelligence knowing at a given instant of time all forces acting in nature as well as the momentary positions of all things ... would be able to comprehend the motions of the largest bodies of the universe and those of the lightest atoms in one single formula, provided his intellect were powerful enough to subject all data to analysis; to him nothing would be uncertain, both past and future would be present to his eyes.

Pierre-Simon Laplace (1799), Traité de Mécanique Céleste.

Laplace was a mathematician-astronomer commenting on how the laws of motion could in principle predict the positions of all bodies, from atoms to planets, in the universe. Scientists generalised this into the creed that everything could be determined, once the appropriate scientific laws had been formulated.

This would, in due course, encompass human behaviour, so that our apparent freedom of will or autonomy would be shown to be illusory:

There was once an old man who said, "Damn!
It is borne in upon me I am
An engine that moves
In determinate grooves,
I'm not even a bus, I'm a tram."

Maurice Evan Hare (1905), Limerick.

All theory is against the freedom of the will; all experience for it.

Samuel Johnson (1778), quoted in James Boswell, The Life of Samuel Johnson (1791).

Autonomous man is a device used to explain what we cannot explain in any other way. He has been constructed from our ignorance, and as our understanding increases, the very stuff of which he is composed vanishes.

B.F. Skinner (1972), Beyond Freedom and Dignity, New York: Knopf.

More recent scientific theories, such as the uncertainty principle (which considers that it is impossible simultaneously to determine the position and momentum of elementary particles) and chaos theory (which is concerned

with how systems governed by physical laws can undergo transitions to highly irregular forms of behaviour) indicate that this naive pessimism in the power of science was misplaced. Although it is quite hard to see how it follows from the fact that it isn't possible to say where an electron is that we must have responsibility for our decision-making, it is perhaps reassuring that the extreme form of determinism, as expressed by Laplace, seems not to hold. At least, we may continue in our belief in free will, as seems to be necessary for our self-esteem:

We must believe in free will. We have no choice.

Isaac B. Singer.

The Yiddish author Isaac Singer (1904-1991) apparently used to offer this comment when pressed for his philosophy of life.

A conventional computer program is entirely deterministic: we require a program which calculates, say, the interest on our bank account consistently to provide the same answer, given the same starting conditions. As we saw, a Turing machine, considered to be equivalent to any computational device, is deterministic: once set in motion it will always perform the same sequence of steps. However, an AI program differs in at least two ways which may fundamentally affect the simple argument that 'Humans are computers; computers are deterministic; therefore humans are deterministic', or, conversely, 'Computers are deterministic; humans are not deterministic; therefore humans are not computers'.

First, it is debatable whether the content of a program should be considered fixed in advance and then just executed. We have seen examples of programs which change themselves during execution (or learn) and hence which it is very hard, even in principle, to say have deterministic behaviour. Even if we disregard learning machines, it is clear that a computer's program can be replaced at will, which is arguably superior to a human, whose program in the form of genes is fixed for the duration of his existence.

Secondly, a Turing machine takes no account of interaction with the environment, that is, with the real world or with users of the machine. These external inputs, not known at the time the program is specified, are not predictable, unless a strong form of determinism holds, and therefore the sequence of operations of the machine cannot be pre-determined. As we have seen, there are profound arguments that aspects of intelligence, such as knowledge, cannot be simply embedded in a program; they emerge only through interaction. So far, there is no formal model of computation, analogous to Turing machines, applicable to interactive machines.

These fundamental limitations may suggest that the Turing machine is not the right theoretical basis for developing the philosophical analogy

between humans and computers. Some of the early proponents of functionalism, which compares the functional properties of the human brain with those of a computer, later withdrew their proposals as they realised that the Turing machine, with its neglect of the environment, was inadequate as a model for human psychology, although, of course, all real computers have sensors and effectors which interact with the world, even if the Turing machine does not:

> **Many years ago, I published a series of papers in which I proposed a model of the mind which became widely known under the name 'functionalism'. According to this model, psychological states … are simply 'computational states' of the brain. The proper way to think of the brain is as a digital computer. Our psychology is to be described as the software of the computer – its 'functional organization'. According to the version of functionalism that I originally proposed, mental states can be defined in terms of Turing machine states and loadings of the memory. I later rejected this account on the ground that such a literal Turing machine-ism would not give a perspicuous representation of the psychology of human beings and animals … We cannot individuate concepts and beliefs without reference to the *environment*. Meanings aren't 'in the head'.**
>
> *Hilary Putnam (1989), Representation and Reality, Cambridge, Mass.: MIT Press.*

Hilary Putnam is not the first person writing about AI to undergo such an apostasy: it may be significant that all such conversions seem to be away from the symbol-processing hypothesis. However, to serve him right, he is still quoted, for example, by Pinker, as an advocate of the computational theory of mind.

Most of the criticisms of the mind-as-computer theory have been directed towards the symbol-processing model, in which there are imagined to be discrete inner entities corresponding to the knowledge structures of computational models. The philosophical implications of connectionist models are unclear. Such models have little structure, which makes it difficult to make sense of and explain those models. The knowledge that they have is distributed throughout the network. It may well be that knowledge and belief are abstractions that one may apply to a complete system but that do not correspond to any identifiable component of that system, much as physical concepts such as energy and inertia can be attributed to a body but cannot be physically pinpointed within that body. As connectionist models do not contain functionally discrete internal states they might provide a different approach to the mind-body problem that is still within the scope of computational functionalism.

53. Consciousness: "the last bastion"

Perhaps even more fundamental than mind is the nature of consciousness itself, a topic to be considered with some trepidation:

> **Science's biggest mystery is the nature of consciousness. It is not that we possess bad or imperfect theories of human awareness; we simply have no such theories at all.**
>
> *Nick Herbert (1985), Quantum Reality, Garden City, N.Y.: Anchor Press.*
>
> **Consciousness appears to be the last bastion of occult properties, epiphenomena, immeasurable subjective states – in short, the one area of mind best left to the philosophers, who are welcome to it.**
>
> *Daniel Dennett (1978), Brainstorms: Philosophical Essays on Mind and Psychology, Montgomery, Vermont: Bradford Books.*

Being a philosopher himself (in fact, professor of philosophy at Tufts University), Dennett took on the challenge and in due course produced the book *Consciousness Explained* (1991). However, his explanation seems not to have been complete for there has been a rapid growth in the number of papers, journals and conferences addressing the issue of consciousness, with (in addition to philosophers) psychologists, biologists, physicists, and even AIers considering that they have a crucial contribution to make to the debate.

In fact, last bastion that it may be, consciousness was one of the very first targets for AI, as discussed at a workshop that pre-dates even the Dartmouth meeting of 1956:

> **The purpose of robotology is to take a hard problem such as this one of consciousness … so that a mixed team can be truly scientific in their work on them. Robotology, then, is a way of solving the communication problem in the sense that we don't just let people talk philosophy, or methodology, or just plain hot air; they must talk in terms of something to be put into the design of an object.**
>
> *Merrill Flood (1951), Report on a seminar on Organizational Science, Report P-7857, The Rand Corporation, Santa Monica, California.*

AI's contribution, then, is supposed to be, as it is with other difficult concepts, to synthesise, rather than just analyse.

The debate about consciousness is heated partly because the concept is itself hopelessly muddled and partly because the various viewpoints are evolving, or perhaps revolving: philosophers of mind consider it to be a matter of psychology; psychologists seek explanations in terms of neurobiology; biology is to be explained in terms of atomic physics; and physicists, because of the uncertainties of quantum physics, are becoming

more philosophical. As far as AI is concerned, its contribution is encouraged, in a manner that is becoming familiar, by the argument that when a concept is very difficult to define, and hence it is hard to decide whether it may be attributed to humans and machines, then we may as well re-define the concept so that it encompasses the machine form as well:

> **It seems preferable to me to extend our concepts so that robots *are* conscious – for discrimination based on the 'softness' or 'hardness' of the body parts of a synthetic 'organism' seems as silly as discriminatory treatment of humans on the basis of skin colour.**
>
> *Hilary Putnam (1975), Robots: machines or artificially created life, Mind, Language and Reality, Philosophical Papers Vol. 2.*

Naturally, many people consider that this begs the very question that is under debate. In any case, the denial of consciousness to robots is made on grounds other than their hardness.

Let us retreat first to the lay understanding of terms such as subconscious, self-conscious, and unconscious, all of which are in vernacular use. Today we are familiar with the idea of the subconscious – that there is a level of the mind's working of which we are not actively aware – but two centuries ago the notion of subconscious thinking would have been regarded as incoherent nonsense. Sigmund Freud (1856-1939) used subconscious mental processes, which he considered to have been repressed and belonging to other selves, the Ego, Id and Superego, within the psyche, to explain previously inexplicable behaviour:

> **All our '56 models have the 'memory' feature but if you can hold until '57 they will have a preconscious and an id.**
>
> *Saturday Review (May 5 1956), Computer salesman in a cartoon by Alan Dunn.*

The fact that there are subconscious mental processes that the individual himself cannot give an introspective account of was, as we saw, soon found to be a difficulty in the attempts to develop psychology through the method of introspection. Indeed, it is a disconcerting thought that an outside observer may have better access to the workings of a mind than the owner of the mind itself.

The term 'self-conscious', in everyday use, refers to the having of knowledge or understanding of oneself, often with a negative connotation, that this self-awareness is somewhat excessive and impedes natural (subconscious) performance. The term 'unconscious' is perhaps the most straightforward, being a medical condition which trained doctors can pronounce upon: it is presumably important for doctors to be able to say if a boxer or someone injured in an accident is unconscious. Of course, there are borderline mental states such as being in a trance, sleeping, hibernating, and

so on, to complicate matters but perhaps the simplest way into consciousness is to consider that 'being conscious' is the medical state of 'not being unconscious'.

So, first of all, we might consider consciousness to be a biological phenomenon:

> **Consciousness is the by-product of the evolutionary process by which our brains integrate our various sensory inputs to give us our vision of the world in which we exist … The integrating area is probably at the base of the brain in the reticular substance because destruction of this area leads to permanent loss of consciousness.**
>
> *Joseph Abrahamson (1994), Mind, evolution, and computers, AI Magazine, 15, 1, 19-22.*

However, nothing can be a *by-product* of evolution, because there is no intended product of evolution. Consciousness is a *product* of evolution, or not, as the case may be. In fact, the explanations of Dennett, Pinker, Edelman and others rely on the argument that Darwin's theory explains the complexity of the mind as well as of the body. Indeed, evolution could be a global explanation of the way anything at all is. But in that case it would be hard to imagine what kind of evidence could falsify it, making evolution what Popper would consider non-science. However, Pinker, at least, does not use evolution to explain everything. Rather inconsistently, he introduces a dualism of his own, separating the mind-brain from what he calls ethics, by which moral values can override mind-brain decisions. But if ethics and morals are so useful to humans, have they not evolved too?

Since consciousness is a property of the human body, specifically of the human brain, it amazes some neurobiologists that it is possible (for AIers, in particular) to address the issue without knowing much about biology. Edelman, for example, points out that AIers are mistaken in:

> **… the notion that the whole enterprise [of AI] can proceed by studying behavior, mental performance and competence, and language under the assumptions of functionalism without first understanding the underlying biology.**
>
> *Gerald Edelman (1992), Bright Air, Brilliant Fire: On the Matter of the Mind, New York: Basic Books.*

Of course, AIers are used to biologists, philosophers, physicists, and so on expressing astonishment that AI is attempted without a deep understanding of biology, philosophy, physics, and so on, respectively, but in order to respond it is necessary to consider what these "assumptions of functionalism" are supposed to be. Roughly speaking, functionalism claims that if the functional roles of aspects of mentality are reproduced, then consciousness

necessarily emerges. Strong versions of functionalism claim that the corresponding computer programs are isomorphic to the processes and capabilities of human thought, but it isn't necessary to take the strong view in order to consider the nature of consciousness. Thus, consciousness, in so far as it is a useful concept to consider existing (and some AIers would deny this), is a kind of epiphenomenon which, because it will emerge from mentality, does not need to be considered directly in the development of thinking machines.

In order to make progress with the ludicrous notion that computer machinery will necessarily have consciousness once they have attained a sufficient degree of mentality we have to try to disentangle some of the knots in the consciousness muddle. Some commentators concentrate on the more private, intrinsic aspects of consciousness, such as pain, anxiety, joy, and so on. For them, the main problem is that of explaining how physical events in the body and brain engender these private experiences. While there have been philosophical discussions about when it would be reasonable to say that a computer is experiencing, for example, pain, on the whole these are not the aspects of consciousness with which AI is most concerned, although, of course, physicists and biologists will argue that if they are neglected then other aspects of consciousness would be impossible. In simple terms, AI is concerned with an agent's awareness of the world and of itself. A distinction is sometimes drawn between 'primary consciousness', which is an awareness of things in the world, and 'higher-order consciousness', which adds a sense of an individual with a past and a future, with humans, of course, being considered to have both.

With humans, the various aspects of consciousness are said to be 'intentional', which means that mental states, such as beliefs and desires, are directed at or about something. Most mind-brain theories and most of AI are concerned with mental states that are, or should be, intentional. According to John Searle, a philosopher critical of the AI programme, intentionality is the key notion in determining whether an entity may be said to have consciousness and whether a computer can be said to think:

> **Because the formal symbol manipulations by themselves don't have any intentionality; they are quite meaningless; they aren't even symbol manipulations, since the symbols don't symbolize anything … Such intentionality as computers appear to have is solely in the minds of those who program them and those who use them.**
>
> *John Searle (1980), Minds, brains, and programs, Behavioral and Brain Sciences, 3, 417-424.*

[According to Searle] consciousness involves a special form of intentionality, namely, that thoughts and words must be 'about' something. For example, an idea can be 'about' a rock, but a rock is not 'about' anything.

George Reeke, (1991), review of Marvin Minsky (1986), The Society of Mind, in Artificial Intelligence, 48, 341-348.

To say that a rock does not have intentions does not help us much: it probably does not have pretensions, contentions, extensions, or retentions, either. We need to know what it is that makes a human brain intentional but a computer not, if that is the case. It is mere sophistry to say that it is because of the 'causal properties' of neurons, as Searle is inclined to do.

Consider the fact that some birds feign injury to distract would-be predators from their nest and that some monkeys give false alarm calls to distract rivals. It is easy to give intentional descriptions of these activities, in terms of desires, beliefs, plans, and so on. It is also easy to see that the behaviours could have evolved because of the advantages they confer. In this case, because the behaviours seem 'automatic', we are inclined towards the evolutionary explanation. Other animal behaviour, such as a dog 'wanting' to go for a walk, is more difficult to describe without the ascription of intentionality. Whether this is in our minds or the dog's is hard to tell, but if we meet an angry dog it matters little.

With human behaviour, it is virtually impossible not to assume intentionality, or so we like to think. However, it is possible to see that consciousness, in the sense of our awareness of ourselves, our past and our future, also confers its evolutionary advantages. For example, if our intentionality provides 'intra-awareness', that is, some awareness of our own mental processes, then it may be adapted to provide 'inter-awareness', that is, some awareness of, or at least intuitions about, the mental processes of others, enabling us to determine, to some extent, the likely reactions to our own actions. The advantages of using simulations to make decisions in economics and other social sciences, as opposed to trying them out in the real world, are so clear that it is possible that consciousness has evolved to enable us similarly to base decisions on mental simulations.

Searle was convinced that thinking implies intentionality and consciousness and that these are not possible in a device that can only shuffle symbols about, as he considered computers to be. According to him, we only make metaphorical attributions of consciousness to our tools, although some AIers aggravated him further by insisting that it made sense to say that devices such as thermostats are conscious. To make his case, Searle imagined incarcerating himself in the sparsely furnished 'Chinese room', which

contained only some paper with instructions detailing how to respond to Chinese messages posted through the letterbox. The instructions would tell him how to shuffle the Chinese symbols to create messages to post back out. Although the instructions would be so good that people outside would be convinced that the input messages were really understood, when he emerged from the imagined room Searle would be certain that neither he, the room nor the paper had the foggiest idea what was going on and that there was nothing corresponding to thinking, intelligence, intentionality, or consciousness.

This experience prompted a multitude of replies, counter-replies, and so on. It is clear that a huge mountain (not a 'few bits', as Searle described them) of paper would be needed, and the person would need remarkable abilities to match and manipulate Chinese symbols in a reasonable time to perform this miraculous feat: but does that make any difference to the philosophical argument? The hypothesis is that the person-paper-room system could answer all Chinese questions, could assert that it understood everything, and could insist that it possessed consciousness, and all the evidence would support this. So the system as a whole seems to understand and has emergent properties that individual components of the system do not have. But then human consciousness too might well be an emergent property, as Searle accepts in his *The Rediscovery of the Mind* (1992).

If consciousness can emerge from arrangements of neurons, why can it not emerge from arrangements of computational elements, if, by hypothesis, they have the same system-level properties? It is not an unfamiliar observation that systems have properties that cannot be said to be possessed by any of its components:

> **The highest activities of consciousness have their origins in physical occurrences of the brain just as the loveliest melodies are not too sublime to be expressed by notes.**
>
> *W. Somerset Maugham (1902), in A Writer's Notebook (1949), Oxford: Heinemann.*

Music has emergent properties that the individual notes do not have; substances have properties that their component molecules and atoms do not have; and so on. Biologists used to believe in a 'life-force' to explain the properties of living matter - until artificial urea, identical to the real form, was created. Similarly, some AIers consider intentions to be mythical:

> **Brentano-Searle intentions are unscientific myths. Despite their seeming immanence, they simply don't exist.**
>
> *Marvin Minsky (1991), Society of Mind: a response to four reviews, Artificial Intelligence, 48, 371-396.*

According to Minsky, the argument that symbol-processing machines (as opposed to neural, connectionist mechanisms) could never be conscious is exactly wrong: it is because our brains are so connectionist (as opposed to symbol-processing) that we have so little consciousness. However, Minsky is mainly considering consciousness in terms of self-awareness.

Others remain more concerned with the physical level:

> **We can, in principle, explain all … input-output performance in terms of activity of neuronal circuits; and consequently, consciousness seems to be absolutely unnecessary! … as neurophysiologists we simply have no use for consciousness in our attempt to explain how the nervous system works.**
>
> *John Eccles (1964), cited in Roger Sperry (1987), Consciousness and causality, in Richard Gregory (ed.), The Oxford Companion to the Mind, Oxford: Oxford University Press.*

Still, many people are reluctant to agree that we don't need the concept of consciousness or that it can be explained away as an emergent property. Such speculations are dismissed as naive and misguided. For example, Roger Penrose considers the notion that consciousness is related to computation to be unbelievable and obviously wrong:

> **Consciousness seems to me to be such an important phenomenon that I simply cannot believe that it is something just 'accidentally' conjured up by a complicated computation ... It is indeed 'obvious' that the conscious mind cannot work like a computer.**
>
> *Roger Penrose (1989), The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics, Oxford: Oxford University Press.*

Well, in that case we are obliged to ask what the conscious mind does work like, although it would take us outside the scope of a book on AI. Penrose's answer appears to be that consciousness can only be understood in terms of some as yet unachieved advances in quantum gravity theory.

Perhaps quantum gravity theory will show that this discussion is becoming too heavy for the average coffee table to support. All the discussion of the various philosophical isms should not be taken to imply that AI system designers themselves are burdened by the philosophical conundrums raised by observers and commentators on their work. Most AIers adopt an engineering stance, being interested most in the performance of systems, be they aimed at practical application or psychological simulation. If an ism is needed for the nature of computing practice, then perhaps the most appropriate is pragmatism:

> **The operative philosophy of the computer world is not logical positivism, or even analytic philosophy, but liberal pragmatism … The successful computer developers and executives no longer talk approvingly (if they**

ever did) of formality, order, hierarchy, and rule, so much as freedom, community, and engagement.

Richard Coyne (1995), Designing Information Technology in the Postmodern Age, Cambridge, Mass.: MIT Press.

As a philosophical school of thought, pragmatism emphasises the practicalities of human involvement and, unlike analytic philosophy, considers theory to be just a form of practice, not superior to it. In so far as system designers need philosophical support, they might find it in the pragmatism of John Dewey (1859-1952) and Martin Heidegger (1889-1976). The pragmatic orientation emphasises the human practices within which a system is to be situated and consequently stresses the social nature of human activities, rather than the independent reasoning of individuals that analytic philosophy focuses upon.

Consistent with the consideration of the social context, pragmatism itself is not asocial. It tends to be associated with a view of computer technology as a liberating, democratic force, enabling users to develop and adapt technology to their own ends, in contrast to a view, which might be seen as derived from analytic philosophy, that computer systems are designed by experts following a systematic process and then delivered, as certifiably correct, to users. Pragmatists are basically optimistic of the roles of technology in society, unlike some anti-rationalists, who consider that technological thinking, which is supposed to separate modes of thinking from its content and context, inevitably leads to decontextualisation, indifference and domination.

54. Applications of AI: "interesting but irrelevant"

With the pragmatist's optimism let us return to more practical matters. Is it not reasonable to expect that what has passed for AI research should bring some tangible benefits to society at large?:

[AI] has now been actively studied for several years and has attracted the interest and support of some extremely able men … At this particular moment in time, is it reasonable to divert intellectual resource into an enterprise which, if I may presume to say so, has been disappointingly slow to produce significant results? There is no doubt at all of the rapid development of computers and of their associated software, but is this enterprise of yours part of the main stream development of computers, or is it merely an interesting, but irrelevant, side line?

Lord Bowden of Chesterfield (1973), Preface to Bernard Meltzer and Donald Michie (eds.), Machine Intelligence, Vol. 7, New York: Halsted Press.

The gentle chiding of Lord Bowden, who as Vivian Bowden had written *Faster than Thought* (1953), one of the first books extolling the merits of computers, shows that AI had already gained a reputation for being "disappointingly slow" in producing significant results by 1973, not yet two decades since the term had been invented. This disappointment was engendered mainly by the high expectations raised by AIers themselves.

Foremost among these expectation-raisers were Allen Newell and Herbert Simon, as we saw earlier. They were also among the first to react to society's unreasonable demand that there should be useful outcomes from AI research by mounting an academic high horse and reminding society that it is confused about the way that science contributes to the enhancement of society:

> **We build computers and programs for many reasons. We build them to serve society and as tools for carrying out the economic tasks of society. But as basic scientists we build machines and programs as a way of discovering new phenomena and analyzing phenomena we already know about. Society often becomes confused about this, believing that computers and programs are to be constructed only for the economic use that can be made of them ... It needs to understand that the phenomena surrounding computers are deep and obscure, requiring much experimentation to assess their nature. It needs to understand that, as in any science, the gains that accrue from such experimentation and understanding pay off in the permanent acquisition of new techniques; and that it is these techniques that will create the instruments to help society in achieving its goals.**
>
> *Allen Newell and Herbert Simon (1976), Computer science as empirical inquiry: symbols and search, Communications of the ACM, 19, 113-126.*

This plea for patience in expecting useful outcomes from studies of the "deep and obscure" phenomena acknowledged by Newell and Simon was made in the paper written as a result of their Alan Turing Award of 1975.

So, if it is not AI's role directly to contribute to achieving society's goals but to develop understanding and techniques that will indirectly do so, we may ask about the nature of such techniques:

> **I think that many of the famous accomplishments in AI are benign kludges, that is to say, I don't think that you can extract from them, successful as some of them are, any systematic deep fundamental science.**
>
> *J. Alan Robinson (1983), Logic programming: past, present and future, New Generation Computing, 1, 107-124.*

Possibly, Robinson considered his own resolution principle to be an accomplishment of mathematical logic rather than AI. Robinson's point is that a successful AI program is a very complicated construction, much of

which is not based on any principled theory. Many programming hacks are necessary to get the program to hang together and these may be more important for its apparent success than any components derived from a theory.

It is not necessary to adopt an adversarial position, in which computers and AI either provide revolutionary solutions to all problems in society or in which they have no effect whatsoever on addressing society's problems. We may modestly claim that they will 'contribute positively', in some unspecified way, towards finding such solutions:

> **Paramount among the problems facing humanity are the control of population, the equitable distribution of wealth, balancing individual freedom against the authority of government, preservation of the world's resources, and the peaceful resolution of conflicts. It is naive to think that there is a primarily technical solution to any of these multifaceted problems … It is not necessary to maintain that computers will have a major role in all of the problems. Through their use in government administration and in the distribution of social services, computers enter into the solution of the major problems and affect the way these problems are attacked. Our belief then, without dismissing present and potential dangers, is that computers have already contributed and will continue to contribute positively toward the solution of difficult social problems.**
>
> *Calvin Gotlieb and Allan Borodin (1973), Social Issues in Computing, New York: Academic Press.*

With this reassuring perspective, we can consider how AI might help in addressing the problems of society.

In the general social services, long before the existence of AI, commentators were prepared to speculate on the benefits of applying mechanical and computational aids to solving problems in health, education, the law, defence, religion, and so on. For example:

> **In the clinics and hospitals of the near future we may quite reasonably expect that the doctors will delegate all the preliminary work of diagnosis to machine operators as they now leave the taking of a temperature to a nurse. Such machine work may be only a registration of symptoms; but I can conceive machines which would sort out combinations of symptoms and deliver a card stating the diagnosis and treatment according to rule.**
>
> *George Bernard Shaw (1918), English Review.*

In other words, Shaw was predicting the use of expert systems for medical diagnosis in doctors' waiting rooms.

It is possible to argue that such a facility is possible today, although it is not generally available, for a variety of reasons, most of them non-technical.

One of the most well known expert systems, MYCIN, to diagnose bacterial infections, which occupied perhaps 100 people-years of design from 1972, was found to outperform general practitioners and to be comparable to experts, but it is not in practical use, and neither are any similar systems. One reason is that the issue of legal responsibility has not been resolved for computer-based diagnosis – an issue that has become increasingly controversial through the provision of medical resources on the internet. A more recent medical expert system, LifeCode, which analyses free-text clinical records of a patient to create an entry for an electronic medical database, is self-aware in the sense that it recognises the limits of its own competence and seeks human assistance if it reaches them.

The dangers of assigning undue faith to any mechanical process, with the consequent suspension of our own critical faculties, had long been recognised, if frequently ignored:

> **He apologizes on his knees for once having postulated in view of a paradoxical result: "although it seems to contradict reality, we must trust our computations more than our good senses."**
>
> *Voltaire (1752), Diatribe du docteur Akakia.*

Voltaire, pseudonym of François-Marie Arouet (1694-1778), was the greatest writer of the Enlightenment, a movement stressing the importance of reason and therefore resisting its delegation to machines or mechanical processes. However, it is not a straightforward conclusion that humans must always retain responsibility over machines. In some cases, where a mechanical aid has been shown to be more reliable than humans, for example, with automatic pilots or delicate surgical operations, it may become illegal to *not* relinquish control to the mechanical aid.

Misgivings notwithstanding, AI has found, as the following sections indicate, many and varied applications. For a definitive assessment of AI's practical contribution we may defer to the ultimate measure of such things, the US Patents Office. During the 1990s the number of patents mentioning AI increased almost twenty-fold, to about two thousand. But this may be only the tip of the iceberg, for:

> **Successful applications of AI are part of, and buried in, larger systems that probably do not carry the label AI inside.**
>
> *Bruce Buchanan and Sam Uthurusamy (1999), The Innovative Applications of Artificial Intelligence Conference, AI Magazine, 20, 1, 11-12.*

So there may be many thousands of patents for larger systems with no explicit mention of its hidden AI.

55. Expert systems: "may help"

Expert systems were the first systems in AI to aim for practical success but it is hard to tell today precisely how successful they really are. Estimates of the number in use range from a handful to a few thousand. Some expert systems are no doubt confidential and unpublished but the main reason for the lack of clarity is that the information technology industry permanently promises new dawns and avoids terms redolent of past decades. Also, the definition of an expert system has become blurred. As often happens, some of the characteristics of expert systems have been merged into more conventional systems, without designers considering the resultant systems to be expert systems:

> **There has developed a consensus that expert systems are not emulations of expertise, but tools to support experts as they go about their familiar tasks.**
>
> *Robert Hoffman, Paul Feltovich and Kenneth Ford (1997), A general conceptual framework for conceiving of expertise and expert systems, in Expertise in Context, Menlo Park, Calif.: AAAI Press.*

In other words, expert systems have evolved away from the idea of a know-it-all expert-in-a-box that replaces human experts towards systems that help and advise experts, by complementing their knowledge of social context with further specialist knowledge.

In the heyday of expert systems, most were aimed towards providing some industrial or commercial benefit but there was a hope that they would also be applied to solving significant problems in society:

> **Expert systems may help. Probably they will. However, we might now be creating a new kind of social monster whose conceit is that all problems may be solved by using the logic and algorithms of a clever chess player ... There is always the danger that people, forever seeking simple solutions, will accept the outputs of machine intelligence with much greater faith than is warranted.**
>
> *Tom Stonier (1984), The knowledge industry, in Richard Forsyth (ed.), Expert Systems: Principles and Case Studies, London: Chapman and Hall.*

Deference to technology is widespread. For example, tennis players are less inclined to berate automatic line judges than human ones. Moreover, human line judges see little point in even making judgements if a machine may contradict them. How much more likely this is with a complex system, when the problem area, for example, economics or medicine, is poorly understood, when the computational technology is incomprehensible, and when there is an appearance of great accuracy.

The search for simple solutions to another social problem, namely, education, had been mocked long ago:

> **I was at the mathematical school, where the master taught his pupils after a method scarce imaginable to us in Europe. The proposition and demonstration were fairly written on a thin wafer, with ink composed of cephalick tincture. This the student was to swallow upon a fasting stomach, and for three days following eat nothing but bread and water. As the wafer digested, the tincture mounted to his brain, bearing the proposition along with it.**
>
> *Jonathan Swift (1726), Travels into Several Remote Nations of the World 'By Lemuel Gulliver'.*

During the 1950s behaviourists developed teaching machines with which students were supposed to learn, like pigeons, by having their behaviour shaped by the reinforcement of acts that tended towards the desired behaviour.

In the 1980s, the imminent success of expert system projects in transferring knowledge from human experts to computer systems led to a hope that 'knowledge transfer' could happen in the other direction, that is, from systems to students:

> **The goal of the Guidon project is to extend the transfer of expertise theme in yet another direction – from the program to the student ... We can expect knowledge-based tutoring to flourish in fields like programming, electronics and areas of science and engineering where there are already established theories ... While we may be already identifying limits to the knowledge-based modeling methodology, we should remember that the advantages of the approach have barely been exploited. The important limits today are clearly practical, not theoretical.**
>
> *William Clancey (1987), Knowledge-Based Tutoring: the GUIDON program, Cambridge, Mass.: MIT Press.*

Although Clancey himself went on to conclude that the important limits were indeed theoretical, as well as practical - for he became an advocate of situated cognition (described earlier) rather than the objectivist philosophy of expert systems he had previously espoused - the search for technological fixes to educational problems continues.

Today the proposition that learners will inevitably acquire knowledge through being given access to the huge volumes of 'information' on the World Wide Web is a cornerstone of many governments' educational policies. All the discussions of information, knowledge and intelligence within a machine inevitably provoke speculations of educational revolutions:

Computers could do what is now unthinkable, and could do it rapidly. They could effectively wipe out illiteracy in the nation.

Frederick Bennett (1999), Computers as Tutors: Solving the Crisis in Education, Sarasota: Faben, Inc.

The argument seems to be that since computers can 'read' and 'write', then surely they can also teach how to read and write.

However, discussion of applications to health and education overlooks the fact that AI research is not generally funded for such purposes, a fact which allegedly led to some difficulty at the MIT laboratory of Seymour Papert and Marvin Minsky. Most AI research has been funded in the United States by the Defense Advanced Research Projects Agency, encouraged, no doubt, by comments on the impact of computer technologies on the nature of conflict:

The computer revolution transforms war into a contest of information rather than of brute force. It enables small cheap devices with brains to overwhelm big expensive vehicles.

Freeman Dyson (1984), Weapons and Hope, New York: Harper & Row.

The mathematical physicist Freeman Dyson won the 1996 Lewis Thomas Prize, which recognises scientists "whose voice and vision can tell us of science's aesthetic and philosophical dimensions, who gives us not merely new information but cause for reflection, even revelation as in a poem or painting". His *Weapons and Hope* would certainly cause AIers to reflect, reinforcing the view that they needed to promise "cheap devices with brains" if they hoped to obtain funding from the defence agencies:

The so-called smart weapons of 1982, for all their sophisticated modern electronics, are really just extremely complex wind-up toys compared to the weapons systems that will be possible in a decade if intelligent information processing systems are applied to the defense problems of the 1990s.

Edward Feigenbaum and Pamela McCorduck (1983), The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World, Reading, Mass.: Addison-Wesley.

Indeed these are not mere promises: the defence agency itself considers that its use of an AI-based logistics planner during the Gulf war of 1991 alone repaid thirty years of investment in AI. Soon, apparently, technology will change the nature of warfare by eliminating the need for humans to be directly involved at all:

The Pentagon, energized by successes in Afghanistan, is moving ever closer to draining the human drama from the battlefield and replacing it with a ballet of machines. Rapid advances in technology have brought an array

of sensors, vehicles and weapons that can be operated by remote control or are totally autonomous.

James Dao and Andrew Revkin (2002), The New York Times, April 16.

It certainly sounds more civilised to replace battles with ballets.

If we are going to let smart machines fight our wars for us, then perhaps they had better pray for us too:

"In fifteen or twenty years' time we shall be writing programs for praying. The subjects and sentiments tend to come in a fairly limited range."

"Ah," said Rowe, "there's a difference between a man and a machine when it comes to praying."

"Aye. The machine would do it better. It wouldn't pray for things it oughtn't to pray for, and its thoughts wouldn't wander."

"Y-e-e-s. But the computer saying the words wouldn't be the same."

"Oh, I don't know. If the words 'O Lord, bless the Queen and her Ministers' are going to produce any tangible effects on the Government it can't matter who or what says them, can it?"

"Y-e-e-s, I see that. But if a man says the words he *means* them."

"So does the computer. Or at any rate, it would take a damned complicated computer to say the words *without* meaning them."

Michael Frayn (1965), The Tin Men, London: Collins.

The relationship between AI and religion is fraught. One line of criticism of the AI endeavour is that it is fundamentally blasphemous, violating the second commandment of Christianity ("thou shall not make to thyself any graven image, nor the likeness of any thing that is in heaven above, or in the earth beneath, or in the water under the earth"). The tension between organised religion and science, deriving from their different notions of truth, is particularly acute in the case of AI, with its inroads into religion's domains of mortality, community, consciousness and the soul. So far, however, theologians, unlike psychologists and philosophers, have not found any inspiration in AI – to the contrary, in fact. On the other hand, if AI is really to address such issues perhaps it needs help from religion, as recognised at MIT, always on any bandwagon in case it should begin to roll, where a theological advisor was added to the robot project team.

Perhaps we could settle for some minor practical roles that AI could take over without objection:

A robot 'preacher' has restored falling attendance at a Sunday School in Nottingham, England, and is now helping the minister with his adult church services as well ... The Rev. Ronald McKenzie, 38 – an electronic technician turned clergyman – constructed the four-foot-nine-inch

mechanical preacher to assist him in his work. "The children are so fascinated that they sit listening to him telling Bible stories while I am out of the room getting on with other work," he said.

SIGART (Special Interest Group on Artificial Intelligence) Newsletter (1974), p44.

As with expert systems, a robot could take over mundane matters, such as preaching to children and parishioners, leaving the human expert to focus on more important matters, whatever they may be.

56. Robots: "must obey the orders given"

The first attempt to specify some commandments for robots had been made by Isaac Asimov in 1942 in his famous laws of robotics:

1. A robot may not injure a human being, or, through inaction, allow a human being to come to harm.

2. A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.

3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

Isaac Asimov (1942), in Runaround, a story first published in the March edition of Astounding Science Fiction, and reprinted in I, Robot (1950).

The laws, specified before computers and certainly before AI existed, place robots in the role of slaves to humans, which is hardly what is needed for autonomous, sentient, conscious beings, if that is what robots of the future will be. Even as they stand, they assume that robots are able to determine when an order will lead to harm to humans. If they are capable of that degree of reasoning, they would probably be capable of determining for themselves what actions to take, without waiting for orders from humans. How do the laws stand up if, as AI proponents expect, robots become capable of decision-making on a par with, or even surpassing that of, humans?

The desire to develop humanoid robots has inspired researchers to, among other things, inaugurate an annual Robotic Soccer World Championships in order to focus research on developing cooperation between autonomous agents in dynamic multi-agent environments and encourage work which "spans the gamut of intelligent robotics research, materials science, electronics, and novel technologies that we cannot even imagine today", which sounds impressive enough to deflect any thought that this is not serious science:

The Robot World Cup Initiative (RoboCup) is an attempt to promote artificial intelligence, robotics, and other related research fields by setting a challenging goal, which is: By 2050, a team of fully autonomous

humanoid robots shall win a game of soccer against the human World Cup Champions, under the official regulation of FIFA.

Hiroaki Kitano (1999), Preface, Special Issue on RoboCup, Artificial Intelligence, 110, 189-191.

This goal is apparently considered reasonable because "it was only 40 years from the invention of the digital computer to the first chess computer that beat a human world champion". Actually, it was 50 years, but what's a decade here or there? Given that almost all the competitors are from Japan and the United States, it is not clear whether the reason for aiming to become world champion at soccer, rather than, say, sumo wrestling or baseball, is technical, academic or socio-political.

This long-term objective is perhaps put into perspective by the Grand Challenge 2004, organised by the US defence department. This required robots, not necessarily of humanoid form and therefore more like unmanned vehicles, to race 150 miles through the Mojave Desert to win a $1 million prize. Of the 106 entries, only 13 made it to the start line, and none got further than seven miles. A consolation is that most soccer pitches have gentler terrain than the Mojave.

Although we have said that developments in computer hardware do not fundamentally affect the achievement of AI, it is clearly the case that they influence research directions and practical applications. The continuing remarkable increases in computing power and decreases in computer size, along with the developments in high bandwidth wireless communication, mean that it is increasingly possible to imagine mobile, autonomous robots with on-board computers capable of intelligent processing in real-time, simultaneously able to interact with other agents (human and computer) in the environment, as would be required indeed by a robot soccer team. In fact, it is no longer necessary to imagine this, for there are prototypes already in service, for example, as urban search and rescuers (for example, in the World Trade Centre after September 11 2001) and as interactive museum guides:

The robot guided thousands of users to exhibits with almost perfect reliability, traversing more than 18.6 km at an average speed of 35 cm/sec. We did not modify the museum in any way that would facilitate the robot's operation; in fact, RHINO's software successfully coped with various 'invisible' obstacles and the large number of people.

Wolfram Burgard et al (1999), Experiences with an interactive museum tour-guide robot, Artificial Intelligence, 114, 3-55.

The robot RHINO is said to have successfully acted as a guide for six days in the Deutsches Museum Bonn.

A more challenging test arena for autonomous robots is that zenith of human intellectual achievement, participation in an AI conference. The GRACE robot was given the following subtasks: navigate to the registration desk; register; interact with other conference attendees; perform volunteer tasks, such as delivering objects to rooms; get to the conference room; give a brief presentation and answer questions:

GRACE successfully completed each of the subtasks described earlier with a minimal amount of extraneous human intervention.

Reid Simmons and 20 co-workers (2003), GRACE: an autonomous robot for the AAAI Robot Challenge, AI Magazine, 24, 2, 51-72.

While the list of subtasks may omit some essential conference activities, they should not be under-rated. For example, registering involves locating the desk, distinguishing people from other objects, finding a queue, checking it's the right queue, getting to the correct end of the queue, standing near (not too close to, not too far from) the last person in the queue, shuffling forward when necessary, chatting with co-queuers if necessary, interacting with the person(s) on the desk, carrying away an immense load of paperwork – all activities that have flummoxed me on occasion. But when the day comes that robots can beat us at soccer and attend our conferences for us, what will there be left to live for?

More mundanely but perhaps more importantly, there are now several hundred robots in US hospitals delivering drugs, food trays and laboratory specimens. The robots are independently mobile in the sense that they plan a route between known destinations in the hospital and use sonar and infrared sensors to avoid obstacles. They open elevator and ward doors, and speak standard messages. Typically, they weigh in at 400 pounds and are about 4 foot 8 inches (1.4m) tall.

The word 'robot' has not yet been refined to distinguish between the various kinds of machines it currently describes. These hospital robots are delivery machines with navigational abilities. They have no decision-making responsibilities (they cannot, for example, decide what drugs to deliver); they cannot use language to interact with people; they do not move like people; they do not get any better at their job; and so on. Some of these capabilities may not matter in this context, as the robots are designed only to perform routine tasks for which there is a labour shortage. Nonetheless, the Japan Robot Association estimates that the robot industry could grow to $22 billion by 2010, although it is likely that, as today, over 90% of robots will be industrial ones engaged on tasks such as welding and painting that require little AI.

A humanoid robot is presumably more of a challenge than a robotic form of other animals. The film, AI – Artificial Intelligence, distributed in 2001, concerned a robot child seeking to develop an emotional relationship with a human mother. The film was quite implausible in suggesting that we might attain this level of artificially intelligent life without various intermediate forms of AI also being widespread in the environment. We are already familiar with or can readily imagine various devices being given attributes which we might consider manifestations of intelligence, for example, cars which can plan a journey, refrigerators which can order food, washing machines which automatically adapt to their washing load, and so on. It seems inevitable that there will be a gradual development in the intelligent functions of man-made devices before the arrival of a robot child.

57. Artificial life: "more alien"

Just as the perceived apex of intelligence has evolved through lowlier life forms, we might expect that artificial intelligence will be developed through lowlier artificial life forms. Perhaps we can start with the lowly worm:

> **Electronic calculators can solve problems which the man who made them cannot solve, but no government-subsidized commission of engineers and physicists could create a worm.**
>
> *Joseph Wood Krutch (1949), The Twelve Seasons, New York: William Sloane Associates.*

On the other hand, some people consider that the level of computer development we have reached is precisely that of a worm:

> **Today, our fastest, most complex computer, armed with our most sophisticated software, is about as complex as a flatworm. Yet, with its explosive self-improvement, how long will it take for the flatworm to become a fish? If we can teach it to adapt on its own, how long will it be before it becomes as complex as we are? If it's changing us so much now, what will happen to us then? Today the computer is a blind, deaf, mute, unfeeling flatworm. One day, though, perhaps sooner than you think, it may walk among us. It will be more alien than anything we can ever imagine.**
>
> *Gregory Rawlins (1998), Slaves of the Machine: the Quickening of Computer Technology, Cambridge, Mass.: MIT Press.*

How long indeed will it take an artificial flatworm to evolve beyond a fish, beyond a human? In a later section, we will see what has been predicted.

The subject of 'artificial life', as with artificial intelligence, requires clarification of what precisely we mean by the natural form. Can we define

the characteristics of life sufficiently that we may agree whether we have created an artificial life form? Arguing that a living being is distinguished by having genes (that is, a set of instructions that enable the being to sustain and reproduce itself) and metabolism (that is, a mechanism to carry out the instructions), Stephen Hawking concludes that:

> **... computer viruses should count as life. Maybe it says something about human nature that the only form of life we have created so far is purely destructive.**
>
> *Stephen Hawking (2000), Life in the Universe, http://www.hawking.org.uk/text/public/life.html*

This definition clearly excludes many forms of artificial animal that have been created since antiquity. For example, three millennia ago mechanical birds, activated by the rising sun, imitated the sound and movement of real birds, and wooden horses, worked by springs, carried out horse-like movements. The most celebrated artificial animal, the duck of Jacques de Vaucanson, built in 1738, could move like a duck and also eat, digest and excrete with sickening verisimilitude. However, its output was some appropriately smelly faecal fake, unrelated to what was apparently ingested. This early lesson that you cannot judge a machine from its output alone was not taken to heart by evaluators of AI. Even today, or perhaps, *especially* today, we are inclined to attribute life-like qualities to our devices:

> **The health and status of the rover is … unknown, but since initiating its onboard backup operations plan a month ago, the rover is probably circling the vicinity of the lander, attempting to communicate with it.**
>
> *NASA (1997), Release 97-255, Mars Pathfinder winds down after phenomenal mission.*

Such descriptions must be considered to be metaphorical, for the devices have nothing corresponding to genes and metabolism, as the definition requires.

Biorobotics, that is, the study of artificial life in the form of animal-like robots, is partly motivated by the need to provide biologists with a means of developing theories of animal behaviour. Just as we might regard AI as a form of theoretical psychology in which the traditional methods to *analyse* human behaviour are complemented by methods to *synthesise* human-like behaviour with computers, so biorobotics complements the analytic methods of biological science with attempts to synthesise life-like behaviours. The recent emphasis on the collaborative or collective nature of intelligence has also led researchers in biology, neuroscience, AI, robotics and graphics to develop models of social insects, such as ants and bees.

If this scientific rationale is unconvincing then it is hard to see a practical justification for creating artificial animals. Not many animals have practical

roles (sheepdogs and Santa Claus's reindeers come to mind) and the natural form is generally adequately abundant. Perhaps artificial sheepdogs and reindeers will be easier to train and cheaper to maintain but this is unlikely to justify the research effort. Maybe a demand for artificial animals will come precisely for those animals that do not have a practical role but serve only as pets. The qualities that people find attractive in pets could perhaps by provided by technology within robotic pets:

> **The pet robot should sound random, purring, chortling, when it is not doing anything significant, and quietly systematic when it is doing something ... [Also] people like to know which way is up on their pets.**
>
> *Robert Rossum (1977), Robots as household pets, Interface Age, 2, 5, 32-37.*

Robert Rossum was the collective nom-de-plume for members of the United States Robotic Society and was a reference to the Rossum of Karel Capek's play *Rossum's Universal Robots*. Such a paper could be written as a joke in 1977 but today robot pets are with us. Sony introduced AIBO, a dog-like machine offered as the first robot pet, in June 1999. Research is being carried out to see if robot pets have a role as a companion for the elderly, in particular. Believing that such pets will form a substitute for grandchildren living far away, the Japan Robot Association predicts that the growing, greying population will sustain a robot pet market of $10 billion by 2010.

58. Cyberocracy: "a state of political Nirvana"

Robert Rossum conceded that some people might find the prospect of robot pets objectionable but argued that "this should not discourage us from pursuing robotics – many people are repelled by cats and dogs". By a similar argument, we might seek to build robotic lawyers and politicians:

> **Could a computer be built, capable of weighing legal evidence, making it possible to circumvent the factor of human error? The idea is not so farfetched as it might appear. The computer has already been introduced into the administrative machinery of the income tax, where it has taken over some of the functions of judge, jury, and executioner. Fearful as we may be of its merciless analysis, we must admit its impartiality. Extension of this idea leads me to suggest the coinage of a new word, cybernocracy, meaning government by computer. Imagine, if you can, a computer sitting in the White House, optimizing the political well-being of each of us. We would continue to vote every four years – not for one program as opposed to another, but for or against a change of programming, and whether to the right or to the left. All of which leads logically to a fascinating concept that might be called differential cybernocracy: a state of political Nirvana**

where change is always possible but revolution impossible – unless someone pulls the plug on the computer!

B.D. Thomas (1965), Science and Society: a Symposium.

The plug-pulling act has played a crucial part in many science fiction novels. In the film *2001: A Space Odyssey*, the on-board computer attempts to take over control of the spaceship:

"I'm sorry, Dave, but in accordance with special subroutine C1435-dash-4, quote, when the crew are dead or incapacitated, the onboard computer must assume control, unquote. I must, therefore, overrule your authority, since you are not in any condition to exercise it intelligently."

HAL, in Arthur C. Clarke (1968), 2001: A Space Odyssey.

Dave's only solution to this situation is to pull the plug on HAL, a last resort that we might imagine will always be open to humans, even though, if the computer really is a sentient, conscious being, then the act might be considered one of murder. As far as cybernocracy goes, we still have the question of who has the right to pull the plug.

Actually, 'cybernocracy' (with an 'n' as in cybernetics) has lost out to 'cyberocracy' (with no 'n' as in cyberspace). Cyberocracy, which is concerned with the impact of information technologies on government, is being increasingly discussed, as it becomes clear that the internet provides new possibilities for distributing and accessing information and for enabling governmental processes, such as voting. However, having an all-powerful decision-making computer in the White House is not imaginable, because no president would so delegate power in the foreseeable future. Even if cyberocracy were able to deliver perfect government, humans would be loath to accept mechanised decision-making from which they have been excluded:

When a machine begins to run without human aid, it is time to scrap it – whether it be a factory or a government.

Alexander Chase (1966), Perspectives.

The implications of a computer role in governmental decision-making were explored in an almost unknown film, *The Forbin Project*, released in 1969. Here, a computer was empowered to decide whether to launch a nuclear missile strike because humans were considered to be unable to evaluate in the time available the complex factors involved. It turned out that the rival power had a similar computer. Sure enough, the two computers learned to communicate and decided to take control of the two countries' missile systems and hold the human race to ransom: they would end war, disease and poverty if humanity submitted to their will.

Of course, these speculations on the roles that advanced computers might play in society have always been accompanied by a rearguard querying the inevitability of technological intrusion into such fields:

> **Questioning the beneficence of scientific rationality and technological progress is almost as heretical as denigrating patriotism ... The belief in the social necessity and inevitability of computer utilities, databanks, management information systems, and sundry computer applications is not based on reason alone. It is the reflection of a political faith built into the scheme of modern history, with an internal logic akin to that portrayed in the Theatre of the Absurd.**
>
> *Abbe Mowshowitz (1976), The Conquest of Will: Information Processing in Human Affairs, Reading, Mass.: Addison-Wesley.*

Abbe Mowshowitz's *The Conquest of Will* was probably the first book to look in detail at the ethics of computer innovations. He considered that the function of all information processing systems lies on a spectrum from the "coordination of diversity" to the "control of disorder" and that only if the human will remained is it possible to prevent the slide to the control end of the spectrum. He warned that the complexity of computerization encouraged the tendency to delegate rationality to the system.

Contemporaneously, Joseph Weizenbaum published the more widely cited *Computer Power and Human Reason*. This focussed more on AI applications, considering many of them unethical or, in Weizenbaum's terms, obscene.

More broadly, social philosophers, especially advocates of so-called Critical Theory, have belaboured the role of technology in society. Their view is not surprising, for nobody attuned to technology is likely to have become a social philosopher in the first place. 'Instrumental reason' – that is, reason that deals with the relation between means and ends but not with the determination of ends – is considered to lead to the 'scientisation' of politics and decision-making – that is, where technical control takes precedence over social goals. Herbert Marcuse, one of the founders of Critical Theory, argued that the technological attitude generates a:

> **… mechanics of uniformity [whereby] individuals are stripped of their individuality, not by external compulsion, but by the very rationality under which they live.**
>
> *Herbert Marcuse (1941), Some Social Implications of Modern Technology, in Technology, War and Fascism, Collected Papers of Herbert Marcuse (1998), edited by Douglas Kellner, New York: Routledge and Kegan Paul.*

Hence, technology may be partly responsible for the Third Reich. Today, these critiques resonate with those who consider that modern technology produces oppressive forms of social control, neglecting human values.

Be that as it may, it is clear that the adjectives 'instrumental' and 'technological' are used pejoratively, to encapsulate a form of reason or attitude that is considered harmful. It is not necessarily the case that social philosophers have studied modern technology sufficiently to be sure that it is sensible to consider that these attitudes may have been somehow generated by that technology, or that the technology somehow exhibits the forms of reason and attitude being so described.

Largely impervious to such critiques, AIers have sought to increase the applied value of their research and today would be able to respond to Lord Bowden's polite question (given a few sections earlier) by quoting a range of practically useful and relatively uncontentious applications of AI. For example: expert systems used by banks to analyse payment histories to detect possible fraud; planning systems that help schedule the maintenance of space shuttle systems; an automated scheduling system that organises the work of Norwegian State Railways employees; speech recognition systems used by travel agents to deal with customer queries; the Google News summary that is generated from the analysis of thousands of news stories every fifteen minutes; image interpretation systems to analyse data collected by airplanes and satellites (including to detect military movements on the ground – pirouettes, perhaps); case-based reasoning systems to give advice on grasshopper infestations, which apparently cause $400m a year losses; an expert system in use in Korea since 1997 to schedule aircraft parking (taking 20 seconds to do what took human experts 4 to 5 hours); and so on. In terms of commercial, industrial and engineering applications, then, AI no longer needs to be apologetically defensive.

59. The future of AI: "rather appalling"

It is time now for us to put disputes and speculations aside and to try to commit ourselves to some assessment of how AI will develop and the contributions it will make to society. It is now nearly fifty years since the subject began as a serious topic of scientific research. It has attracted the professional attention of computer scientists, psychologists, philosophers, mathematicians, physicists, and biologists, as well as the fascination of the general population. Many projects have been pursued, ranging from the theoretical to the applied. What now, on the basis of the accumulated experience, is our assessment and what would we predict?

Predicting the future is always difficult. Wise people refrain:

I never make predictions and I never will.

Paul Gascoigne, footballer and philosopher.

However, AI is inherently concerned with the future and AIers feel obliged to make predictions. The pioneer Alan Turing had, even before AI research had begun, anticipated one of the difficulties of making and assessing such predictions, namely that the meanings of words are not fixed for all time. The meanings evolve as words are used in different contexts:

At the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted.

Alan Turing (1950), Computing machinery and intelligence, Mind, 56, 236, 433-460.

This phenomenon is particularly marked in a fast changing field such as computing, where new techniques and concepts need new words to describe them. Rather than confuse people with newly invented jargon (for which computing is notorious), old words are sometimes adopted and used in an initially metaphorical fashion, although the metaphor is soon forgotten, again leaving outsiders confused. We can see this with the use of everyday words such as web, browser, cache, link, chat, page, and so on in internet terminology. Is there any evidence that the predicted evolution in the meaning of the specific words 'machine' and 'thinking' has occurred?

As is often the case, contemporary advertisements provide an insight into the general view of such concepts. Many advertisements rely for their effect on a kind of post-modern understanding that thinking is something that we have been encouraged to believe that machines can do but which in reality they cannot. For example, we see robot car painters doodling until they observe a human supervisor approaching. The impact of such an image would be lost if the viewer really believed that robots were able to behave in such a fashion. If, today, the word 'thinking' is used with reference to machines it is more often than not intended to be ironic. On the other hand, the word 'intelligent' is freely used to describe various products, most of which are flattered by the term. So, it seems that Turing's predicted evolution has not yet fully occurred and that, on the whole, the terms 'machine' and 'thinking' are still considered to be mutually incompatible. In any case, whether machines can be said to think or not is a question of linguistic usage: it doesn't say anything about the actual capabilities of machines.

With the passing mention of post-modernism, it is perhaps worth saying that deconstructivists would not consider propositions such as 'machines can think' and 'computers are intelligent' to be ones whose truth or falsity can be

established by defining conditions, such as passing the Turing test, that would enable the propositions to be asserted to be true. Instead, the truth of any such proposition (say, 'Marvin Minsky is a luminary', for a change) lies in what the proposition reveals and conceals, and in what thoughts, taking account of the context in which the statement is being interpreted, are prompted by this metaphorical use of language, which all language use is, according to deconstructivists – a view supported by psychological studies which indicate that metaphorical uses require no more processing than apparently literal uses. Similarly, meaning is said to lie not in truth-values but in the interplay between differences in points of view, the aim of thinking being not to reach a conclusion as in rationalistic thinking but to keep ideas in play. Regardless of whether such notions of truth and meaning are useful, it is clear that the proposition 'machines can think' has provoked a great many thoughts and differences of viewpoints, with no conclusion yet being reached.

As regards AI, we have seen that opinions have oscillated from euphoria to gloom. Let us begin at the depths of despair, at the time when large amounts of money were being lost on the expert systems 'boom':

> **AI has a rather appalling future as a slogan rationalising the greed-driven triumph of hope over expectations.**
>
> *William Janeway (1984), Financing the future, in Patrick Winston and Karen Prendergast (eds.), The AI Business, Cambridge: MIT Press.*

At least, from here a balanced view will seem relatively positive.

60. Past predictions: "far from realization"

As we come to contemplate today's predictions, we might reflect upon past predictions of a future time that has now in fact passed. As we re-read such predictions today, we should bear in mind that all researchers are obliged to emphasise the potential significance of their research: it is the main way to gain research funding. On the other hand, an unrealistic over-emphasis on potential outcomes will inevitably cause problems for researchers and their colleagues in due course. Therefore, we have to assume that these quoted predictions, which were made, after all, by leading AIers themselves and not by the fringe enthusiasts that AI has always attracted, represent a considered, if optimistic, view of their opinions at that time.

The current generation of AIers will wince at the resurrection of one of the earliest and most notorious of such predictions made by Herbert Simon and Allen Newell just two years after the Dartmouth meeting that launched the subject of AI:

> **There are now in the world machines that think, that learn and that create. Moreover, their ability to do these things is going to increase rapidly until – in the visible future – the range of problems they can handle will be coextensive with the range to which the human mind has been applied ... Within ten years a digital computer will be the world's chess champion, unless the rules bar it from competition ... will discover and prove an important new mathematical theorem ... [and] most theories in psychology will take the form of computer programs, or of qualitative statements about the characteristics of computer programs.**
>
> *Herbert Simon and Allen Newell (1958), Heuristic problem solving: the next advance in operations research, Operations Research, 6, 1-10.*

Simon and Newell's predictions are important today because of the evidence they provide of the zeitgeist of the early AI period. The details are less important now but clearly, with the hindsight of decades beyond when the predictions should have come to pass, they under-estimated the difficulty of the three tasks they mentioned.

As we have seen, the world's chess champion is still not quite a computer. The situation is becoming as confused as the world boxing championships, with its multiple boards and their own champions. The Kasparov – Deep Blue match of 1997 was played under conditions that arguably favoured Deep Blue. For example, Kasparov was not able to see any previous games of the version of Deep Blue that played the match. The Kramnik – Deep Fritz match of 2002 was not a championship as Deep Fritz was not the reigning world computer chess champion. Moreover, Kramnik was allowed to experiment with Fritz before the match (the Fritz program had to be frozen for several months to permit this) and to consult Fritz during a game if it was adjourned. The draw might therefore be considered to indicate computer superiority, although an eight game match is short by championship standards. The Kasparov – Junior match of January 2003, involved the world No. 2 (although his chess rating is higher than Kramnik's) and the chess No. 1, a title won in July 2002. It ended with one win apiece and four drawn games.

What constitutes an "important new mathematical theorem" is perhaps a matter of opinion. One of the most significant theorems to be first proved by a computer is the famous four-colour theorem, which states that if a map on a flat surface is to be coloured so that no two countries that share a common border have the same colour then this can be achieved without using more than four different colours. A computer program developed by Kenneth Appel and Wolfgang Haken proved this in 1976. Their experience with their

program sheds another light on the 'computers can only do what they are programmed to do' adage:

> **When we had hand-checked the analyses produced by the early versions of the program, we were always able to predict their course, but now the computer was acting like a chess-playing machine. It was working out compound strategies based on all the tricks it had been taught, and the new approaches were often much cleverer than those we would have tried. In a sense the program was demonstrating superiority not only in the mechanical parts of the task but in some intellectual areas as well.**
>
> *Kenneth Appel and Wolfgang Haken (1977), The solution of the four-color-map problem, Scientific American, 237, 4, 108-121.*

The most difficult open mathematical problem so far to be solved by a computer is apparently the Robbins algebra problem, an accomplishment featured in the New York Times in December 1996. However, these are not proofs of a 'new theorem': they are new proofs of old theorems, and some mathematicians remain unsure that they count as proofs. Certainly, the theorems were not 'discovered' by a computer program.

Simon and Newell's third prediction is the least precise and therefore the hardest to deny categorically but a cursory examination of any textbook presentation of theories in psychology would show that most of them are not, in fact, presented as computer programs.

In 1964, Arthur Samuel, mentioned earlier as the person who developed a checkers program that learned to outplay him, made a series of predictions about the situation that would exist twenty years later, in 1984:

> **... programming as we know it will have ceased to exist ... communication with a computer will then be easy and natural ... it will be possible to dial anywhere in the world and to converse with anyone speaking a different language ... libraries for books will have ceased to exist in the more advanced countries ... paper-work will cease to exist ... the working week will have been shortened to four days ... the world draughts, chess and go champions will, of course, have met defeat at the hands of a computer ... computers will have largely taken over the task of composing and arranging music ... [but] all attempts to invest them with truly creative abilities will have failed.**
>
> *Arthur Samuel (1964), The banishment of paperwork, New Scientist, Feb. 27, 529-530.*

Even now, the only one of these nine predictions that can reasonably be said to have taken place is the last one, which coincidentally is the only one predicting what computers would *not* achieve, although, to be fair, his other predictions concerning developments in computer hardware were virtually

spot on. It is worth reflecting on some of these predictions to try to see how our visions may be so astray.

The issue of whether or not the general population will become computer programmers has always been controversial. John McCarthy, another member of the Dartmouth committee of 1956, believed the opposite to Samuel:

> **It may be supposed that, as happened with television and then color television, the enthusiasts and the well-to-do will be the first to install computer consoles in their homes. Eventually, however, everyone will consider them to be essential household equipment. People will soon become discontented with the 'canned' programs available; they will want to write their own. The ability to write a computer program will become as widespread as the ability to drive a car.**
>
> *John McCarthy (1966), Information, Scientific American, 215, 3, 64-223.*

The word 'programming' is perhaps another which has evolved so that it is hard to know now whether to agree or disagree with such a prediction. McCarthy's comparison of writing a computer program to driving a car is not one that many would have accepted in 1966. At that time, writing a program was more akin to being a car mechanic, able to design, build and mend the innards of a car engine. Today, however, writing one's own programs is, in fact, more like driving, than engineering, a car. Most programs today are written by the judicious adaptation of existing pre-designed packages, many of them generally available on the internet, rather than by designing a new program from scratch. This established and accepted methodology is only now being applied (similarly supported by the internet) to the task of writing in general, to the consternation of the copyright industry and others concerned about plagiarism.

The impact of technology on working practices, to which Samuel also refers, was, of course, much discussed and continues to be so. The consensus in the 1960s and 1970s was that technology would inevitably lead to a decrease in our working hours:

> **As these technologies progress, we will be seeing much of the drudgery taken out of the life of our citizens, while productivity levels continue to rise. Instead of the predicted gradual decrease in the 40-hour workweek, I believe the exponential growth and uses of our technologies could result in a 20-hour workweek by the 1980's.**
>
> *C. Lester Hogan (1972), As the industry sees it, Communications of the ACM, 15, 510.*
>
> **The bottom 30%, and perhaps as many as 70%, will be among those for whom many sources of employment will have dried up ... They will**

generally be in a lower IQ bracket also ... [but a] solution may emerge allowing the less capable fraction of society not to work at all and to be supported by those who do.

James Martin and Adrian Norman (1970), The Computerised Society, Englewood Cliffs, N.J.: Prentice-Hall.

Computers, computer technology and automated techniques can replace most of the jobs of most of the people for most of the time.

Clive Jenkins and Barrie Sherman (1977), Computers and the Unions, New York: Longmans.

How could we be so wrong? Today it seems that those who are working, which is the greater majority of those who want to, are working harder than they were thirty years ago, even allowing for the fact that it is in their interests to say so. It may well be the case that technology would allow us to work a 20-hour week to sustain a 1972 lifestyle. However, this ignores the nature of a capitalist system. In general, a company that worked a 25-hour week would be more successful than one that worked a 20-hour week, so the 20-hour week would soon become a 0-hour week. At a certain point, however, an extra five hours a week is counter-productive. This point is determined by a cultural consensus, not by what is enabled by technology. In short, technological advance is virtually irrelevant to the hours we work.

Even so, predictions of the decrease of work continue to be made. For example, Hans Moravec predicts that human-like robots will enable our descendants not to work at all:

Inevitably, such a development will lead to a fundamental restructuring of our society. Entire corporations will exist without any human employees or investors at all. Humans will play a pivotal role in formulating the intricate complex of laws that will govern corporate behavior. Ultimately, though, it is likely that our descendants will cease to work in the sense that we do now. They will probably occupy their days with a variety of social, recreational and artistic pursuits, not unlike today's comfortable retirees or the wealthy leisure classes.

Hans Moravec (1999), Rise of the robots, Scientific American, 282, 124-135.

It reminds me of the gardener who, when made redundant by an artificial digger and asked what he would do with all his new leisure time, replied, "I likes to dig."

Apart from changing working practices, computers would also revolutionise other social activities:

It may be possible for intelligent machines of the future to supply not only intellectual stimulation or instruction, but also domestic and health care,

social conversation, entertainment, companionship, and even physical gratification.

Oscar Firschein, Martin Fischler, L.Stephen Coles and Jay Tenenbaum (1973), Forecasting and assessing the impact of artificial intelligence on society, Proceedings of the 3rd International Joint Conference on Artificial Intelligence, 105-120.

Passing quickly over physical gratification, the sociability of intelligent machines (conversation, entertainment and companionship) still leaves something to be desired. Their role in domestic and health care is also minimal.

However, Oscar Firschein and friends were not alone in anticipating changes in education:

We are at the onset of a major revolution in education, a revolution unparalleled since the invention of the printing press ... By the year 2000 the *major* way of learning at all levels, and in almost all subject areas, will be through the interactive use of computers.

Alfred Bork (1979), Interactive learning, American Journal of Physics, 47, 5-10.

This new method of learning would, of course, bring drastic institutional changes, since standard schools and universities could not provide the individualised, self-paced courses needed with interactive learning. Seymour Papert had a rather different vision of how computers would revolutionise education, based on his expectation that learning environments would be re-designed so that computer-based systems would better support constructivist learning activities. Asked in 1975 to predict changes in education that would have occurred in ten years time, he declined – but, like Bork, was prepared to predict them for 2000:

Ten years is in some ways a challenging and in some ways a very awkward period for predicting the impact of computers on education. If you asked me whether the practice of education will have undergone a fundamental change through the impact of computers in either five years or in twenty years, I could answer with complete confidence 'no' to the first question and 'yes' to the second.

Seymour Papert (1977), A learning environment for children, in Robert Seidel and Martin Rubin (eds.), Computers and Communication: Implications for Education, New York: Academic Press.

It is perhaps a matter of opinion as to whether a "fundamental change" has occurred. Certainly, it is not the case that most learning is through interactive computers, as Alfred Bork predicted. In reality, educational institutions have a sensible inertia resisting change and, yet, because of the huge investment

society has made in them, an incentive to evolve towards meeting the new demands.

Similarly, the prediction that libraries and paperwork would soon disappear seems ludicrous now, although Samuel was far from alone in believing it:

> **There is no real question that completely paperless systems will emerge in science and other fields. The only real question is "when will it happen?" We can reasonably expect, I feel, that a rather fully developed electronic information system ... will exist by the year 2000, although it could conceivably come earlier.**
>
> *F. Wilfrid Lancaster (1978), Whither libraries? Or, wither libraries?, College and Research Libraries, 38, 345-357.*

The fact that technology makes something possible does not mean that it will happen. In this case, for example, libraries serve so many functions that it is obvious (now) that they will adapt to provide the newer functions needed, rather than disappear. The advent of the World Wide Web has perhaps provided the "rather fully developed electronic information system" that Lancaster envisaged but it has not led to the demise of the library or the disappearance of paper.

Samuel's prediction that we would be able to speak directly to anyone in the world – implying instantaneous natural language translation – was a remarkably bold one, considering that many of the problems of automatic translation were already well recognised in 1964. Needless to say, it hasn't happened. The prospect of it happening remained sufficiently remote for Douglas Adams to lampoon the suggestion in the *Hitchhiker's Guide to the Galaxy*:

> **The Babel fish is small, yellow and leech-like, and probably the oddest thing in the Universe. It feeds on brainwave energy received not from its own carrier but from those around it ... It then excretes into the mind of its carrier a telepathic matrix formed by combining the conscious thought frequencies with nerve signals picked up from the speech centres of the brain which has supplied them. The practical upshot of all this is that if you stick a Babel fish in your ear you can instantly understand anything said to you in any form of language.**
>
> *Douglas Adams (1986), The Hitchhiker's Guide to the Galaxy, London: Guild Publishing.*

The Babel fish was later honoured by having an automatic translator on the web named after it. In fact, it should be acknowledged that the web translators, although far from perfect, are useful enough to show that progress is being made even if at a slower rate than some predicted.

Twenty years on, Samuel reflected on his own predictions. He conceded that, as far as his predictions about AI were concerned, they were little nearer being fulfilled in 1983 than they had been in 1964. Although happy to make further predictions concerning hardware developments, he resisted the temptation to predict anything about AI:

> **… my 1963 AI predictions are nearly as far from realization today as they were in 1963, and this in spite of the fact that there has been much more research, and good research, done on AI in general, than I envisioned possible in 1963 … [after a set of hardware-related predictions] I am much less sure as to where AI research will be in the year 2000.**
>
> *Arthur Samuel (1983), AI, where it has been and where it is going, Proceedings of the 8th International Joint Conference on Artificial Intelligence, 1152-1157.*

If Samuel had not died in 1990 we might have expected similar comments to be made in 2003 or so.

61. Ultraintelligence: "an intelligence explosion"

In the early days of AI there was a widespread and understandable tendency to extrapolate extravagantly from the apparent initial successes. Here, for example, is a set of predictions that 'ultra-intelligent machines' will soon overcome us:

> **It is unreasonable ... to think machines could become nearly as intelligent as we are and then stop, or to suppose we will always be able to compete with them in wit or wisdom. Whether or not we could retain some sort of control of the machines, assuming that we would want to, the nature of our activities and aspirations would be changed utterly by the presence on earth of intellectually superior beings.**
>
> *Marvin Minsky (1966), Artificial Intelligence, Scientific American, 215, 9, 247-260.*
>
> **If that happens at Stanford, say, the Stanford AI Lab may have immense power all of a sudden. It's not that the United States might take over the world, it's that the Stanford AI Lab might ... Eventually, no matter what we do there'll be artificial intelligence with independent goals. It's very hard to have a machine that's a million times smarter than you as your slave.**
>
> *Edward Fredkin (1979), quoted in Pamela McCorduck, Machines Who Think, New York: W.H. Freeman.*
>
> **Let an ultraintelligent machine be defined as a machine that can far surpass all the intellectual activities of any man however clever ... It is more probable than not that, within the twentieth century, an ultraintelligent**

> **machine will be built and that it will be the last invention that man need make, since it will lead to an "intelligence explosion".**
>
> *Irving John Good (1965), Speculations concerning the first ultraintelligent machine, in Franz Alt and Morris Rubinoff (eds.), Advances in Computers 6, New York: Academic Press.*
>
> **In from three to eight years we will have a machine with the general intelligence of an average human being … At that point the machine will begin to educate itself with fantastic speed. In a few months it will be at genius level and a few months after that its power will be incalculable.**
>
> *Marvin Minsky (1970), Life magazine, November 20.*

Minsky later said that "the Life quote was made up. You can tell it's a joke". Well, it certainly seems laughable now. Unfortunately, people were misled by similar comments and the quotation continues to be given as if it were serious. Indeed, it is similar in spirit to a contemporaneous opinion:

> **Within a generation, I am convinced, few compartments of intellect will remain outside the machine's realm – the problems of creating 'artificial intelligence' will be substantially solved.**
>
> *Marvin Minsky (1967), Computation: Finite and Infinite Machines, Englewood Cliffs, N.J.: Prentice-Hall.*

It is not so easy to dismiss a comment made in a serious, highly-regarded textbook on formal computability as it is one in a light-hearted magazine article.

Why did some AIers think that it was reasonable to speculate about machines "a million times smarter" than we are and to expect machines soon to surpass all our intellectual capabilities? Analogies are always risky but compare how we make predictions about improvements in athletic performance. Consider a new event, such as the women's pole vault, where we are at the stage of initial success. In the first four years of the event, the world record improved from 4.41m to 4.81m. So, after a further fifty years, by a linear extrapolation the record will stand at 9.81m. Perhaps, beyond a certain point, a 'vaulting explosion' will occur, giving a record of 98.1m, say.

What happens in athletics is that performance fairly soon levels off, new records become increasingly rare, and eventually they become so rare that nobody is much interested in them any more. Similarly, for 'artificial IQ' we might expect performance levels to even out, probably well below the human level in the most important kinds of intelligent activity. Obtaining any further improvement would become prohibitively expensive. Researchers and the public would turn to other problems. This is at least as plausible a scenario as one in which machines attain incalculable power within a few months.

By and large, AIers have learned the lesson from the over-ambitious predictions of the first decade or two of AI. However, as we've seen and will see, AI still attracts practitioners who cannot resist presenting their visions of the future. More likely nowadays, though, is that the hubris of the field is channelled into startlingly bold long-term projects, such as, to develop an all-encompassing theory of cognition, to develop an encyclopedic knowledge base with human-like common sense, to develop a team of robots capable of defeating a team of humans at soccer, and so on. These are safer than audacious predictions because at the end of the project the precise original aims may have been forgotten and there are bound to be some successful outcomes along the way.

Even so, even long-term projects come to an end and the critics of AI will be there waiting to hold researchers to account. For example, Dreyfus's 1992 addition to his AI critique is almost entirely based on the anticipated failure of the CYC project, where Lenat placed himself in the position of a sitting duck through his grandiose claims. This expected failure is then gleefully proclaimed as ending the AI dream:

> **It seems highly likely that the rationalist dream of representationalist AI will be over by the end of the century.**
>
> *Hubert Dreyfus (1992), What Computers Still Can't Do, Cambridge, Mass.: MIT Press.*

It is odd that Dreyfus, who has capitalised the most on the premature extrapolations of AIers, should be led to make such a prediction himself. Into the new century, it is clear that representationalism is far from over.

62. The AI industry: "a serious marketplace"

After the first flush of idealistic speculation, realism set in and with the development of expert systems in the 1980s AI became much more hard-nosed. The Japanese government as part of their Fifth Generation Computer Systems project coolly presented the prospect that the practical application of AI techniques would provide the cornerstone of the new information society:

> **In the 1990's when it is expected that fifth generation computer systems will be in wide use, information processing systems will be central tools in all areas of social activity to include economics, industry, art and science, administration, international relations, education, culture and daily life and so forth.**
>
> *Tohru Moto-oka, ed. (1982), Fifth Generation Computer Systems, Amsterdam: North-Holland.*

The reputation of the Japanese for adopting ideas from Western research to reap economic rewards prompted emergency reactions from the United States and Europe.

In the US, the open market made a coordinated government response difficult but a large number of entrepreneurs leapt into action:

> **Artificial intelligence is bursting out of the research lab and into the marketplace ... To serve this expanding market, dozens of startup companies are now being joined by IBM, DEC and other traditional vendors who see the enormous potential of AI applications. AI is no longer a nebulous, "blue sky" dreamworld, but a serious marketplace.**
>
> *Ken Sonenclar (1985), publicity for Gartner Group, Inc.*

The European Community began the ESPRIT research and development programme, which continues to the present day. As far as Americans were concerned, the British had a crucial role to play:

> **Perhaps it is unfair to single out the British failure in artificial intelligence when Britain has done so badly elsewhere in computing too … The only reason for dwelling on the British example is that it demonstrates what it was like to have had everything in place to excel, and yet by mis-management, by misperceptions, by folies de grandeur, and other delusions, the British have demonstrated how to turn a nation from a winner to a loser. In England's tragedy there is an obvious lesson for Americans.**
>
> *Edward Feigenbaum and Pamela McCorduck (1983), The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World, Reading, Mass.: Addison-Wesley.*

In other words, Britain (or England? – they seem to be indistinguishable to Feigenbaum and McCorduck) served as a warning of how it could all go wrong.

So, a fifth generation systems industry was born in Japan, the United States and Europe with the aim of leading a social revolution. What kind of vision of the new society did its proponents have?:

> **... through the intellectualization of these advanced computers, totally new applied fields will be developed, social productivity will be increased, and distortions in values will be eliminated … We are ... seizing this new medium to do better one of the things we've always liked to do best, which is to create, pursue, and exchange knowledge with our fellow creatures. Now we are allowed to do it with greater ease – faster, better, more engagingly, and without the prejudices that often attend face-to-face interaction.**

Edward Feigenbaum and Pamela McCorduck (1983), The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World, Reading, Mass.: Addison-Wesley.

No hint of folies de grandeur and other delusions there then. Presumably, these advanced, 'intellectualized' computers would eliminate distortions of values such as those illustrated by the authors' views of the British. The authors' vision of expert systems seems to be that they will sanitise the distortions and prejudices that characterise our everyday interactions, eliminating the need for face-to-face confrontations.

One prominent member of the British AI community, Donald Michie, the first director of the first European AI lab, in Edinburgh, felt sufficiently uncowed to join in with the predictions that we would soon have to cope with the "torrential outpourings" of "knowledge-foundries":

The torrential outpourings of the knowledge-foundries and skill-synthetics plants of 1999 A.D. will be eminently reinjectable and assimilable by the consumer. If a person wishes to master a new branch of knowledge, or a new aptitude of mind or body, to a level beyond today's professional standards, and to do it fast, and if he is not prevented by limits set by his constitutional capacities (if IQ < 90, then avoid algebraic topology; if limbs < 4 then leave polevault alone, etc.) then he will find ample means to do this. What sort of society will evolve under such conditions? No-one knows. Certainly one can forget about employment; it has not been a concern of any leisured and cultivated class of the past. We have to envisage such a class expanded to include the whole population, and at the same time the attainable quality of leisure and culture expanded upwards.

Donald Michie (1985), in Daniel Bobrow and Patrick Hayes (eds.), Artificial intelligence – where are we?, Artificial Intelligence, 25, 375-415.

It certainly takes a feat of imagination to envisage the whole population, which much prefers soap operas to operas, as being or becoming a leisured and cultivated class, and not many of us can forget about employment.

It did not take long for a reaction to such visions to occur. Corporations that invested large amounts of money on the AI promises of the early 1980s lost heavily. So, after a brief glimpse of spring, the 'AI winter' set in. It has proven to be more of an ice age than a winter, for it is still the case that any mention of artificial intelligence is particularly foolhardy if made by anyone seeking financial backing for a computer-based innovation.

If AI is considered to be concerned with that which hasn't yet been programmed then the notion of an AI product is rather self-contradictory. If there were such a product it would be better marketed under some other description and its designers better labelled with some name other than AI

expert (and in fact that is happening: many companies have employees who, in other times, would have been considered to be developing AI applications). Still, it was possible for enthusiasts to claim the birth of an 'AI industry' in the 1980s:

Overall, the industry went from a few million in sales in 1980 to $2 billion in 1988.

Stuart Russell and Peter Norvig (1995), Artificial Intelligence: A Modern Approach, Englewood Cliffs, N.J.: Prentice-Hall.

This, rather conveniently, beats the 1984 forecast made in *Electronics Week* of a $2 billion industry by 1989.

The ten-year Fifth Generation Computer Systems project duly came to an end in 1992, with, as tends to be the case with political programmes of this sort, only an implicit review of its successes and failures. The Japanese then moved smoothly on to another ten-year project, called the Real World Computing programme, with a similar $500 million budget. This project, in contrast to its predecessor, which was considered to focus on symbol-based information processing, was intended to pursue the aspect of 'intuitive' or 'pattern-based' information processing in order to provide 'flexible intelligence'. While this might sound like a fundamental research programme, it had in fact mainly applied objectives, being led by a dozen or so Japanese companies. This new program did not provoke the flurry of nervous responses from the West that the Fifth Generation project had done.

In the 1990s, the AI industry was re-invented mainly in terms of agent-based systems. The rapid growth in the discussion of agents began before the existence of the World Wide Web although that has become their natural habitat. Typical applications of agents are as digital personal assistants, intended to help busy people manage interactions in cyberspace much as human assistants might help them in the real world. They would deal with electronic mail, carry out information searches, and prepare reports from electronic data. We can also imagine tasks that we would not normally assign to human assistants, unless they are exceptionally amenable:

If I have a good idea in the middle of the night, I'd like to explain it to my electronic personal assistant; by the time I arrive at work, I'd like the system to have worked on the problem by finding me papers I should read or people I need to talk to. I think 10 years from now we might be able to do some of this.

Howard Shrobe (2000), What does the future hold?, AI Magazine, 21, 4, 41-57.

That would certainly be an improvement on the bedside notepad, which invariably tells us that the middle of the night bright idea is dimmer in the cold light of day.

With the data for many businesses, such as travel agencies, booksellers, estate agencies, banking, libraries, and so on, being increasingly on-line, there is great scope for adding to conventional software that processes these databases various agent-like properties to facilitate activities such as searching, negotiating, collaborating, analysing and learning. In 2002, the Gartner group forecast that by 2012 what they call 'enterprise automation', which includes autonomous software agents and artificial intelligence software to make independent decisions, such as automatically searching for and purchasing products on the web, would be worth $250 billion, which would be almost 50 per cent of total IT. However, there are regular forecasts informing companies of good times ahead that rarely seem to arrive.

Applications such as knowledge-based systems are AI's contribution to the much heralded 'information society', which we are in the process of establishing. The principal asset of this post-industrial society is (or will be) information and knowledge. Although the first steps have been faltering, we must hope that AI plays a full role in such a society because it is clear that we will need advanced techniques to make the best use of the information resources. If it comes to pass as some social commentators expect then an information society will be very different from its predecessors because, for the first time, its main resource can, in principle, be shared at no cost, unlike, for example, man-power, minerals, and capital. Maybe this will contribute towards a global community that is inherently cooperative rather than competitive. At least, we can already see that the flow of information has the potential to be more democratic, thereby preventing control by dictatorial governments.

63. AI as a science: "another scientific revolution"

It is tempting to see the future of AI in terms of increasingly useful applications that will benefit everyday life. For example, we previously considered some early speculations on the prospects for medical expert systems, automated teachers, robot preachers, and so on. It may, however, be the case that the main contribution of AI will be theoretical rather than practical.

For example, in the social sciences many of the problems are concerned with causal reasoning in some form: Will this new drug be beneficial? How should we deal with young offenders? Does the World Wide Web encourage sexual deviance? Will improving the railways raise industrial productivity? And so on. At the moment, social scientists really only have the methodology of statistics to tackle such questions: there is no significant theoretical or

practical basis for dealing with causality. AI, however, has had to address the question of causality, not as a philosophical divertissement but because it aims to build machines that can operate in the world and hence reason about cause and effect. To do so, it has had to develop declarative schemes for describing the world 'as it is' and procedural schemes for analysing how the world may change as the result of various events and actions. Maybe these will provide the theoretical foundation for causal analyses of immense practical benefit in the social sciences, as Judea Pearl, director of the Cognitive Systems Laboratory at the University of California at Los Angeles and winner of the 2001 Lakatos Award of Outstanding Contribution to the Philosophy of Science, has argued:

> **… several hurdles [to handling causal analyses] have recently been removed by techniques that emerged from AI laboratories. I predict that a quiet revolution will take place in the next decade in the way causality is handled in statistics, epidemiology, social science, economics and business … The development of autonomous agents and intelligent robots requires a new type of analysis in which the doing component of science enjoys the benefit of formal mathematics side by side with its observational component … I am convinced that the meeting of these two components will eventually bring about another scientific revolution, perhaps equal in impact to the one that took place during the Renaissance. AI will be a major player in this revolution.**
>
> *Judea Pearl (2002), Reasoning with cause and effect, AI Magazine, 23, 1, 95-111.*

If this predicted revolution in causal analyses in the social sciences does occur, it may be anticipated that its genesis within AI will be forgotten.

Others have identified different aspects of AI research as providing theoretical advances important for the social sciences, for example, the work on multi-agent systems, which is concerned with modelling the behaviour of societies:

> **AI can significantly contribute to solve the main theoretical problem of all the social sciences: the problem of the micro-macro link, the problem of theoretically reconciling individual decisions and utility with the global, collective phenomena and interests. AI will contribute uniquely to solve this crucial problem, because it is able to formally model and to simulate at the same time the individual minds and behaviors, the emerging collective action, structure or effect, and their feedback to shape minds and reproduce themselves.**
>
> *Cristiano Castelfranchi (1998), Modelling social action for AI agents, Artificial Intelligence, 103, 157-182.*

There are, however, great difficulties to be overcome before such revolutions in the social sciences may occur. Social scientists are familiar with their established methodologies and do not have the means to easily understand and adapt AI technologies.

Moreover, there is a constitutional reluctance to accept that computers can really help with social questions:

> **To claim that the computer will ever master our messy human realities or indeed improve the mind's way of dealing with them is ... a sign of the madness of our time.**
>
> *Theodore Roszak (1987), cited in Enid Mumford, Managerial expert systems and organizational change: some critical research issues, in Richard Boland and Rudy Hirschheim (eds.), Critical Issues in Information Systems Research, Chichester: John Wiley & Sons.*

Theodore Roszak, Professor of History at California State University, wrote the wonderfully titled *The Cult of Information: a Neo-Luddite Treatise on High Tech, Artificial Intelligence, and the True Art of Thinking* (1994), which, after a broad swipe at science in general, homes in on AI for forcing the trend towards quantitative, data-based decision-making, to the neglect of more qualitative, intuitive and worthwhile thinking.

This follows a critique of the so-called 'cult of information' that allegedly has an agenda to shift power to the technological elite. AI is said to be wrong to compare a machine's information processing with a human mind's thinking because, according to Roszak, thinking involves 'ideas' and not information. He considers that ideas – especially so-called master ideas, such as "all men are created equal", that are the basis for our morality, spirituality, metaphysics and culture – can only be created and appreciated by humans.

Roszak naturally takes advantage of the open goal that AIers have provided by showing them either to have deluded themselves or to be seeking to delude others in their assessments of intelligent computers:

> **AI has been peculiarly characterized by extravagant, often propagandistic claims in its own behalf, with the result that authorities in the field have contributed as much to the folklore of computers as the advertisers, promising machines that would translate languages, understand speech, process visual images, make legal, political, and financial decisions, and, in general, outstrip human intelligence in every application.**
>
> *Theodore Roszak (1994), The Cult of Information: a Neo-Luddite Treatise on High Tech, Artificial Intelligence, and the True Art of Thinking, Los Angeles: University of California Press.*

As usual, however, Roszak's treatise is undermined by a simplistic view of the nature of computation. Perhaps Roszak found it difficult to devote adequate time to computation whilst writing a series of books on topics as wide-ranging as ecopsychology, sexual mythology, and longevity, as well as several science fiction books, including *The Memoirs of Elizabeth Frankenstein*, the adopted sister of our old friend Frankenstein.

Perhaps he was confused by computer scientists who have, after all, sought to reassure those overwhelmed by technology that computers only transfer bits of information or process symbols and all computations are equivalent to what can be done with a particularly simple machine – but have added that, by the way, computers will soon be more intelligent than humans. It is as if we had just invented music and said, "there is no need to worry, all music is based on just individual notes played together or in sequence – but it can express the profoundest of human emotions". Or we had said that literature was just one word after another, and yet the means for expressing the deepest of human thoughts. Roszak does not consider the situational, interactional view of computation, or the nature of parallel, connectionist computers, or the intentional approach of agent-based methods, or much else that is the concern of modern AI research. Such writings are therefore not so much critiques of AI but indications of the cultural antipathy that needs to be overcome if AI is to have a positive influence on social scientists and humanists.

There are two sides to this issue. It is unreasonable to say that 'soft scientists' are not au fait with AI when AIers have, on the whole, been determined to establish AI as a 'hard science', comparable to physics or engineering. This view of AI is based as much on funding policies and the politics of universities as it is on the nature of the subject. It is easier for AIers to obtain funding if their laboratory is within a hard science department – and they will naturally do research which fits best within such a department. Soft scientists will often feel alienated and indeed unwelcome within such an environment.

Whether or not AI may be successfully applied to problems in industry and society, the science of AI may continue, if researchers remain interested in it. Or it may not, if the experience of AI research leads to the conclusion that the endeavour is futile:

One cannot construct machines that either exhibit or successfully model intelligent behavior.

Terry Winograd and Fernando Flores (1986), Understanding Computers and Cognition: A New Foundation for Design, Norwood, N.J.: Ablex.

Winograd, the creator of what was regarded as one of the most successful models of intelligent behaviour, came to the conclusion that the goal of AI is unattainable. Such a categorical conclusion requires an agreement on the goal of AI – for there appear to be many – and of the meanings of "exhibit", "model", and "intelligent behaviour".

The conclusion of Winograd and Flores followed a critique of the rationalist tradition of the scientific method (although their description of this method hardly satisfied those engaged in it), arguing that it needed to be replaced by a 'phenomenological' philosophy derived principally from Heidegger. It is hard for those immersed in a particular tradition to see the virtues of a proposed different one. At a superficial level it seems that the main problem was considered to be that AI inevitably lacks the context needed for intelligent behaviour, being forever limited to the programmer's representation of the world. Practically, the outcome is a recommendation that we should not attempt to implement intelligent machines but that we should instead design computers to facilitate communication between humans.

At least Winograd and Flores have illuminated for the AI community the rather impenetrable writings of Heidegger, as Hubert Dreyfus had earlier tried to do, on the way that technology distorts authentic thinking. They are to be commended for suspending any scepticism they might have felt about the views of someone who can advocate the importance of contextualising technology, whilst himself remaining aloof from the social and political context within which he worked, that is, 1930s Germany. In so far as the 1986 book was intended to revolutionise or end AI research it would appear to have had little effect, being scarcely referenced in current AI texts. Its influence, in fact, has been significant, if more indirect, with a critique from within being more willingly assimilated by AIers.

As we have seen, many commentators of different backgrounds have argued that AI is misguided and doomed to failure. For example, Roger Penrose did not consider that the AI view of truth and consciousness corresponded to his intuitions as a mathematical physicist. Unfortunately, his argument in *The Emperor's New Clothes* was presented as a mathematical reductio ad absurdum proof when the real difficulties in AI are not susceptible to a traditional mathematical analysis. His follow-up book argued again, via 400 pages of quantum physics, that standard AI could not succeed. His comments directly on AI, however, are superficial. For example, when discussing the progress of chess-playing programs, he says:

> **The human player needs to keep reapplying judgements and forming meaningful plans, with an overall understanding of what the game is all**

> **about. These are qualities that are not available to the computer at all … the essential point is that the quality of human *judgement*, which is based on human *understanding*, is an essential thing that the computer lacks.**
>
> *Roger Penrose (1994), Shadows of the Mind: A Search for the Missing Science of Consciousness, Oxford: Oxford University Press.*

Of course, by definition, a computer lacks *human* judgement and understanding but AIers will wonder in what sense chess programs are not forming judgements and plans. That seems to be precisely what they are doing. And what exactly is a game of chess "all about" that a computer doesn't understand? The argument is misdirected: if Penrose had discussed computer-based diplomacy rather than chess programs then a point may have been conceded.

Until critics demonstrate a detailed understanding of AI methodologies and techniques, most AIers will be reluctant to invest time in familiarising themselves with quantum physics, Heideggerian philosophy, neural Darwinism, and so on. Until then, AIers will conclude that they should be more concerned if experts in other fields did *not* react aggressively to the intrusions on their territories.

64. Future AI machines: "unimaginable"

One or more of the critics may well be right and AI, as it is currently understood, may turn out to be of little long-term significance. In fact, it is possible that the whole area of computer science, which has proliferated throughout the university system, is over-rated:

> **Computers turn out in the end to be rather like cars: objects of inestimable social and political and economic and personal importance, but not the focus of enduring scientific or intellectual enquiry.**
>
> *Brian Cantwell Smith (1996), On the Origin of Objects, Cambridge, Mass.: MIT Press.*

This seems a surprisingly suicidal point of view for someone, Brian Cantwell Smith, who is regarded as one of the leading philosophers of computer science. More importantly, it seems to be wrong, for computers are indeed the focus of profound academic study in their own right and there is hardly an aspect of human intellectual enquiry that has not had to re-consider its foundations at the prospect that machines may replicate that enquiry, in some sense.

As we know, others have been more optimistic about AI. The stage of development reached, and to be reached, by AI continues to be much discussed. In 1998 James Allen, at that time president of the American

Association for Artificial Intelligence (AAAI), drew an analogy between AI and aviation, which he considered to have had an infancy of several thousand years, an adolescence beginning in about 1900 with the first powered flight, and an adulthood from the development of the science of aeronautics and a deep understanding of the nature of flight:

> **I believe that we're at a similar transition point to the first flight because we are now able to construct simple working artifacts which can then be used to support experimental work. This is a critical event in the development of the field that I believe will revolutionize the way the field operates and the way it is perceived.**
>
> *James Allen (1998), AI growing up, AI Magazine, 19, 4, 13-23.*

Although AI has so far had an infancy of only fifty years or so (unless we include pre-computer creations such as the Egyptian oracles or Pascal's calculator) and is now entering its adolescence, its adulthood surely cannot be far away, according to Allen.

Indeed, a few years before, it had already been reached, according to an earlier president of the AAAI:

> **At that point [1980], we had already made it through our infancy, when simple things were exciting and being done for the first time … Then we hit the raging hormones of adolescence … By now, though, we have emerged into the beginning of adulthood.**
>
> *Elaine Rich (1992), Editorial, AI Magazine, 13, 2, 12-13.*

This maturity and respectability is reflected in the professional journals and conferences of AI, which now have a solemn mathematical, scientific style, replete with theorems and data. The lessons of the early, exaggerated expectations have been learned. AI is now proper science, with none of the ridiculous pretensions, so one would think, that the levels of artificial intelligence will soon reach those of human intelligence and will then soar ahead.

Or is it? A few pages back, it was asked how long it would take for an artificial flatworm to evolve beyond a human. Hans Moravec conveniently lays out a timetable for us, anticipating:

> **… four generations of universal robot, each spanning a decade. The first has lizardlike spatial sense, the second adds mouselike adaptability, the third monkeylike imagination, and the fourth humanlike reasoning. The uniform schedule comes from matching each prototype animal's brain to steadily rising computer power.**
>
> *Hans Moravec (1999), Robot: Mere Machine to Transcendent Mind, Oxford: Oxford University Press.*

So, according to Moravec, we will have human-like robots by 2040 or so:

By 2040, I believe, we will finally achieve the original goal of robotics and a thematic mainstay of science fiction: a freely moving machine with the intellectual capabilities of a human being.

Hans Moravec (1999), Rise of the robots, Scientific American, 282, 124-135.

This prediction is based upon extrapolating increases in computer power and comparing these with the brain size of lizards, mice, monkeys and humans, without any real consideration of the concomitant software developments.

It is, however, fairly certain that hardware developments alone cannot deliver the performance required and that the really difficult problems lie in the design of the programs needed, as has always been the case:

The speeds and memory capacities of present computers may be insufficient to simulate many of the higher functions of the human brain, but the major obstacle is not lack of machine capacity, but our inability to write programs taking full advantage of what we have.

John McCarthy, Marvin Minsky, Nathaniel Rochester and Claude Shannon (1955), proposal to the Rockefeller Foundation for the Dartmouth Summer Research Project on Artificial Intelligence.

(The 'may' is charmingly reticent, considering that they were referring to the computers of 1955.) The history of AI itself shows that the phenomenal increases in computer power over the last fifty years have not been matched by increases in their 'brain-power'.

A degree of scepticism is therefore in order, especially as Moravec had previously predicted (in 1988) that we would have "mobile utility robots to help us around the house" by 1998. The Japanese firm Matsushita said in 2002 that, after a $1.5 million development, it hoped to market a vacuum-cleaning robot "in two to three years time". However, Electrolux has beaten them to it, with the Trilobite (named after the extinct marine anthropod), the first robotic vacuum cleaner to go on sale (for about £1000 in 2004). It is considered to be a glimpse of the future rather than an effective tool for today. Perhaps Matsushita will win the race for the other promised home robots, such as security guards or caretakers for children or the elderly, when equipped with features like cameras and mobile connections. The robot child carer will be equipped with eyes in the back of its head.

But Ray Kurzweil considers the 2040 estimate too generous. He thinks that we will have human-level AI soaring past us by 2030:

Three-dimensional molecular computing will provide the hardware for human-level 'strong' AI well before 2030. The more important software insights will be gained in part from the reverse-engineering of the human brain, a process well under way. Once nonbiological intelligence matches

> **the range and subtlety of human intelligence, it will necessarily soar past it because of the continuing acceleration of information-based technologies.**
>
> *Ray Kurzweil (2002), Human-level 'strong' AI: the prospects and implications, abstract for invited talk, 18th National Conference on Artificial Intelligence.*

He is sufficiently confident about this to have bet $10,000 that the Turing test will have been passed by 2029. It is true that there have been major advances in neuroscience in the last decade or two but it does not follow that it will be straightforward to 'reverse-engineer' these discoveries into principles for software development or to directly scan human neural circuitry into neural computers, as Kurzweil suggests.

However, Kurzweil clearly knows many things that we do not, for as the inventor of machines to read, recognise speech, and synthesise music he has received the $500,000 Lemelson-MIT Prize, the US's largest award in invention and innovation, and the 1999 National Medal of Technology, the nation's highest honour in technology. He is also the author of *The Age of Spiritual Machines: When Computers Exceed Human Intelligence* (1999), which, despite its title, is not about computational spirituality, except in the sense that it argues that the inevitable development of computer technology will make it possible, within the present century, for people to become immortal by downloading their minds into computers.

So a child born today may 'live' forever. If what Kurzweil foresees comes to pass, we can dispense with the chore of creating new people, unless we are not content with the billions of minds we will already have. To return to reality, or at least to the more immediate future as predicted by Kurzweil, we have been promised human-level AI by 2030. But now Rodney Brooks, director of the MIT AI Laboratory, thinks that it will all have happened by 2022, with "unimaginable" changes occurring by 2007:

> **Today there is a clear distinction in most people's minds between the robots of science fiction and the machines in their daily lives … Our fantasy machines have syntax and technology. They also have emotions, desires, fears, love, and pride. Our real machines do not. Or so it seems at the dawn of the third millennium. But how will it look a hundred years from now? My thesis is that in just twenty years the boundary between fantasy and reality will be rent asunder. Just five years from now that boundary will be breached in ways that are as unimaginable to most people today as daily use of the World Wide Web was ten years ago.**
>
> *Rodney Brooks (2002), Flesh and Machines: How Robots Will Change Us, Pantheon Books: New York.*

Any retreat on 2022?

Why do AIers persist with such speculations? Brooks, Kurzweil and Moravec are leading members of the AI robot research community and if anybody should be able to make reliable predictions about robot development then they should. However, the large majority of AIers resist the temptation to speculate and are embarrassed and annoyed by the predictions of others. In the early days of a grand mission like AI some romanticised exaggeration was inevitable and welcome (in order to get AI research off the ground) but fifty years on those chickens are well roosted. Today, in order to attract funding, it is necessary to over-egg the pudding a little but there is no magic pudding to which it is possible repeatedly to apply this trick. Bold predictions are certainly more interesting than bland ones and are more likely to be included in collections of quotations. Perhaps they are not bold at all. Time will tell.

65. Impact on the psyche: "men may become robots"

According to some commentators, then, we are still on course for the superior artificial intelligence soberly predicted long ago, surely, by 2034:

> **Many AI scientists believe that artificial intelligence inevitably will equal and surpass human mental abilities – if not in twenty years, then surely in fifty.**
>
> *Nils Nilsson (1984), Artificial intelligence, employment and income, AI Magazine, 4, 2, 5.*

Nils Nilsson, for a long time director of SRI International's AI Centre and now Emeritus Professor of Engineering at Stanford University, has written a series of textbooks laying the academic foundations of AI and is regarded as one of its more restrained and responsible commentators. He is careful to attribute to "many AI scientists", which may or may not have included himself, what must have seemed, in comparison with the predictions of others, the safe upper estimate of fifty years to develop AI that would surpass human intelligence.

He followed the obligation implicit in the opinion that the arrival of artificial intelligence was inevitable to consider the impact that this will have on humanity's view of itself. He considered that the outcomes would be wholly beneficial, because seeing man as a machine would avoid all the suffering caused by the old-fashioned view that man is a spiritual being:

> **The next twenty years will be a period of great rethinking of many legal and philosophical issues. For example, will the agreements entered into by our 'intelligent agents' be binding upon us? – upon them? AI will compel new views of 'man as mechanism'. This time, it is likely to be the whole**

> **man – not just parts of him – that are regarded as machines. My view is that we will find it much more humane to think of man as a machine than as a spiritual being. Far too much mischief and suffering have resulted from the latter view.**
>
> *Nils Nilsson (1985), in Daniel Bobrow and Patrick Hayes (eds.), Artificial intelligence – where are we?, Artificial Intelligence, 25, 375-415.*

It is hardly surprising that theologians have so little enthusiasm for AI if this is the view of one of the more responsible AIers. Herbert Simon more modestly concluded that man would lose his egocentric view of his place in the universe but did not commit himself to the form that the new view would take:

> **The definition of man's uniqueness has always formed the kernel of his cosmological and ethical systems. With Copernicus and Galileo, he ceased to be the species located at the center of the universe, attended by sun and stars. With Darwin, he ceased to be the species created and specially endowed by God with soul and reason. With Freud, he ceased to be the species whose behavior was – potentially – governable by rational mind. As we begin to produce mechanisms that think and learn, he has ceased to be the species uniquely capable of complex, intelligent manipulation of his environment. I am confident that man will, as he has in the past, find a new way of describing his place in the universe – a way that satisfies his needs for dignity and for purpose. But it will be a way as different from the present one as was the Copernican from the Ptolemaic.**
>
> *Herbert Simon (1961), The corporation: will it be managed by machines?, in Melvin Anshen and George Bach (eds.), Management and the Corporations, New York: McGraw-Hill.*

There is at least no denying the chutzpah with which just five years after the phrase 'artificial intelligence' had been invented its practitioners placed themselves in line with Galileo, Darwin and Freud.

If new intelligent life forms come to exist, how will humanity relate to them, especially if, as we have seen has been predicted, they outstrip us in intellectual capability? Will we, like proud parents, take pride in the success of our progeny, rendering ourselves obsolete?:

> **Today, our machines are still simple creations, requiring the parental care and hovering attention of any newborn, hardly worthy of the word "intelligent". But within the next century they will mature into entities as complex as ourselves, and eventually into something transcending everything we know – in whom we can take pride when they refer to themselves as our descendants.**

Hans Moravec (1988), Mind Children: The Future of Robot and Human Intelligence, Cambridge, Mass.: Harvard University Press.

However, intelligent computers would not be descendants of humans in the same sense that humans are descendants of apes. We may talk loosely of humans evolving into computers but this is not the scientific meaning of evolution. Humans design computers. They are no more our evolutionary successors than are aeroplanes or refrigerators.

How will these superior intellects, if they come to exist, regard their predecessors? Will they allow us to be in any position to take pride in their activities? Will they feel any obligation to continue our existence? They might allow us to be preserved, as man has, rather belatedly, come to do with displaced races and endangered animals:

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.

Aaron Sloman (1978), The Computer Revolution in Philosophy: Philosophy, Science and Models of Mind, Brighton: Harvester Press.

Or maybe smart computers will puzzle over whether carbon-based life forms could possibly be intelligent:

Someday computers may laugh at us and wonder if biological information processors could really be smart. Beware of those who think it can never happen. Their ancestors hassled Galileo and ridiculed Darwin.

Patrick Winston (1977), Artificial Intelligence, Menlo Park, CA: Addison-Wesley.

Ah, Galileo and Darwin again. Our ancestors also hassled and ridiculed thousands of other scientists, with names long forgotten, whose ideas did not turn out to be so significant as those of Galileo and Darwin. Naturally, AIers consider that they will belong with the latter set.

Another prospect is that robots will come to regard us as their slaves in a reversal of the roles we tend to see for robots today, that is, as pseudo-intelligent beings capable of, for example, exploring hostile environments in our stead. The interplay between men, robots and slaves is a recurring theme. Erich Fromm, a sociologist who considered freedom the central characteristic of human nature, felt that we had a misperception of the potential danger:

The danger of the past was that men became slaves. The danger of the future is that men may become robots.

Erich Fromm (1955), The Sane Society, New York: Rinehart.

This is, of course, a less optimistic interpretation of Nilsson's prediction that men will view themselves as machines rather than spiritual beings.

In almost all the comments on the Turing test it is assumed that success will mean that the machine's thinking will have improved to become comparable to that of humans. Another perfectly reasonable interpretation is that, by then, human thinking will have deteriorated to become comparable to that of machines. If it seems implausible that such an apparently natural human capability as thinking will change under the influence of machines, consider the case of some dyslexic students given speech recognisers in the expectation that this would improve their linguistic performance. In the event, their use of language deteriorated because they sub-consciously reacted to the fact that speech recognisers are not 100% reliable and used short, simple sentences to avoid the frustration of having long, complex sentences misunderstood. The same thing happens when language translation systems are used. In other words, we automatically adapt to the systems we use and this is even more likely to happen if we ever become regular users of apparently thinking machines.

66. The future of humanity: "All is Machine"

Humans may become more robotic in a straightforward physical sense. Computational implants – for example, to enable us to see better (predicted by 2010 by Rodney Brooks) or to be linked directly to the web via brain implants (by 2030) – may become commonplace:

> **While we have come to rely on our machines over the past 50 years, we are about to become our machines during the first part of this millennium. We need not fear our machines because … those of us alive today, over the course of our lifetimes, will morph ourselves into machines.**
>
> *Rodney Brooks (2002), Flesh and Machines: How Robots Will Change Us, Pantheon Books: New York.*

In this vision, there are not two intelligent species, robots and humans, but only one ('romans', perhaps – although technologists will no doubt prefer 'hubots').

The thought that we may merge with our machines is not a new one:

> **People who spent most of their natural lives riding iron bicycles over the rocky roadsteads of this parish get their personalities mixed up with the personalities of their bicycles as a result of the interchanging of the atoms of each of them and you would be surprised at the number of people in these parts who nearly are half people and half bicycles.**
>
> *Flann O'Brien (1967), The Third Policeman, London: MacGibbon and Kee.*

Likewise, many bicycles were half-human too. Some philosophers have delighted in a literal kind of thought experiment in which components of the

brain are replaced, part-by-part, by functionally equivalent computers, wondering when or if our sense of self would disappear. It is unlikely that they imagine such an operation would be socially acceptable although apparently some roboticists see no fundamental difference between, say, a computer-controlled heart pace-maker and a computer replacement for the human brain.

The term 'cyborg' is usually taken to mean entities that use both organic and inorganic components, such as pace-makers and prostheses: Brooks's vision would also encompass components capable of autonomous, intelligent decision-making. The brain taboo is already being broken by the acceptance of computational neural implants to control diseases such as multiple sclerosis, Parkinson's disease and cerebral palsy.

Even if AI never develops to the point that robots are superior enough to enslave us, it is possible that AI will have a dehumanising, alienating effect on society. The vision of Rodney Brooks, a leading member of the AI community, will undoubtedly provoke reactions of repulsion and revulsion from many people. We seem to feel it necessary to believe that humanity has some unique role and place in the universe and if AI provides yet more evidence that this is not the case, it is likely to have an unsettling effect, at the minimum. Many modern neuroses may derive from the constant undermining of our decision-making, autonomy and responsibility. Some socio-political theorists see AI as intended to have just this effect, as part of authoritarian control, to show that we are just machines and therefore should behave accordingly.

One discussion of the social and legal implications of the advent of artificial intelligence concluded that they were so dire that the ownership of intelligent machines might be banned, an act considered similar to the abolition of slavery:

> **The political and social impact of a societal decision to proscribe ownership of "intelligent machines" would be as great or greater than the impact of the abolishment of slavery.**
>
> *Marshall Willick (1983), Artificial intelligence: some legal approaches and implications, AI Magazine, 4, 2, 5-16.*

It is difficult to see how the ownership of an intelligent machine could be made illegal, since the definition of such an entity is virtually impossible. Many everyday devices, such as cars, cookers, watches, and so on are being endowed with functionalities which some might argue show the glimmerings of intelligence. This is bound to continue, with no clear threshold beyond which the outcomes might be thought harmful to society. Still, it is certain that society will need to try to find one. For example, consider ALVINN, an

autonomous vehicle that uses computational vision to recognise roads and traffic and neural networks to learn how to drive. What would be the legal position when such a vehicle has an accident? We have already reached such a situation with air traffic. As this section was being written, two planes collided over Germany and the cause was immediately said to be either human error or a fault in the air traffic control software.

It may seem premature to speculate on the legal status of intelligent machines, for example, to be concerned over whether they have any legal rights or whether they are responsible for any unfortunate outcomes. On the other hand, societal attitudes do change and perhaps faster than we realise, although perhaps not as fast as some would like. After all, it is not so long ago that women were regarded (by men) as essentially different from men, in terms of their rationality, competence and general legal status.

General consideration of the legal, ethical and social implications of AI has largely been shelved. If there is a public perception of AI it is that it provides a source of entertainment and not much more. AI has a track record of promising computers that could read encyclopedias, communicate in natural language, compose music, clean the house, and so on – most of which has not materialised in the form predicted. Of course, we can say now that the predictions were naive and based on a lack of appreciation of how difficult the scientific problems are. But these were promises and predictions of AI experts, so the field can hardly complain if the public has a sceptical view of the apparently slow progress. This is unfortunate and possibly perilous if it hinders the proper consideration of AI's role in society. Compared to genetic engineering, which some may see as having a similar goal of re-designing the human race, there is little public debate of the implications of AI. Eventually, however, there will have to be a realistic understanding of the actual and potential achievements of AI, if sensible decisions are to be made.

Are there other reasons for believing that the arrival of artificial intelligence is not the inevitability that some commentators seem to believe? One possibility is that we will come to realise that we really don't need artificial people when we have more than enough real ones, nine billion later this century, according to most estimates:

> **As we look around the globe and witness its enormous population, one of the first features that must strike our attention is just how little we need more things that think like human beings ... while the intellectual appeal of creating replicas of human mind in its full range and complexity will be strong, the practical demands will be weak.**
>
> *William Chace (1984), Intelligence, artificial and otherwise, AI Magazine, 5, 4, 22-25.*

It is an easy comment to make that the huge number of human minds, and the relative ease of replenishing the supply, makes the development of artificial minds unnecessary. This, however, is hardly the aim of AI, although, if pushed on the point, it might be argued that human minds, despite their great abundance, have not yet shown the quality needed to deal with the planet's problems – indeed, they have created most of them:

If ever a species needed to be replaced for the good of the planet, we do.

Isaac Asimov (1978), And it will serve us right, People's Computing, 7, 1, 16-20.

A defining characteristic of a 'species' is that its members are capable of interbreeding. The extent to which future AI machines, which is presumably what Asimov had in mind, may be said to interbreed will be a fascinating philosophical and practical enquiry.

The usual justification for any scientific endeavour, regardless of any eventual practical benefit, is that it furthers our knowledge of the world and, especially in the case of AI, deepens our understanding of our own place in that world. Our melancholic review of the many predictions made about the impact and achievements of AI should not be interpreted as providing a Popperian falsification of the AI endeavour. As we have seen, AI is not a single, precisely defined theory, and none of the AI variations leads, in the way that we require of a scientific theory, to the derivation of the predictions that we have discussed. Rather, the predictions are a manifestation of the admirable, if sometimes misplaced, enthusiasm of AIers. The failure of the predictions does not disprove any theory of AI.

It remains the case that since its inception as an academic subject in 1956 or so AI has developed hugely in the depth and breadth of its activities. The technical sophistication of many of its areas is daunting to non-participants. For example, Hubert Dreyfus, AI's perennial critic who is always anxious to argue that AI cannot possibly provide commonsense reasoning, has to admit that he cannot read any of the thousands of AI papers written on the topic. AI has spawned sub-areas, such as computational vision, robotics, planning, learning, and so on, that now have almost separate existences, with their own journals and conferences.

AI's particular methodologies, providing theoretical rigour with a practical orientation, have enlightened many associated disciplines such as philosophy, psychology and computer science. Of course, unexpectedly profound problems have been encountered and it may turn out that AI's contribution to associated disciplines will be less significant than its enthusiasts thought. Nonetheless, AI has not only enlightened them but has also enlivened them because it has revived important but hitherto moribund discussions. At least, it has been shown that we should avoid simplistic

dismissals of AI based on the grounds that, because it seems to be about machines, it must be anti-humanist and therefore pernicious and dangerous:

> **Contrary to what most people assume, this field of research has a potential for counteracting the dehumanizing influence of natural science, for suggesting solutions to many traditional problems in the philosophy of mind, and for illuminating the hidden complexities of human thinking and personal psychology. The common view that machine research must tend to display us humiliatingly to ourselves as "mere clockwork" is false. The more widely this is realized, the less of a threat will artificial intelligence present to humane conceptions of society.**
>
> *Margaret Boden (1977), Artificial Intelligence and Natural Man, New York: MIT Press.*

Through the advent of AI, then, humanity may come, apparently paradoxically, to an enhanced view of itself.

On balance, then, do we concede to the view of Edward Fredkin, with which we started, that the appearance of AI is or will be the last event of consequence for this planet? This is a view not only of the AI avant-garde but also one familiar to science fiction aficionados:

> **First, they were creeping moulds that slithered forth from the ocean onto land, and lived devouring one another, and the more they devoured themselves, the more of them there were, and then they stood upright, supporting their globby substance by means of calcareous scaffolding and finally they built machines. From these protomachines came sentient machines, which begat intelligent machines, which in turn conceived perfect machines, for which it is written that All Is Machine, from atom to Galaxy, and the machine is one and eternal, and thou shalt have no other things before thee!**
>
> *Stanislaw Lem (1974), Prince Ferrix and the Princess Crystal, translated from Polish by Michael Kander, New York: Seabury Press.*

In response to this quasi-religious manifesto, we could deny that intelligent machines have appeared and assert our faith that they never will, thereby making predictions of machine omnipotence irrelevant. We might argue that, even if AI were to be possible, humanity would forever retain ultimate control, by means of the hand on the on-off switch, sufficient to maintain its place on the planet. The step from humanity to machines is far from an evolutionary inevitability. Evolution itself may be inevitable but we are the first species to be challenged by a process that we ourselves have initiated – and that we could (perhaps) control or stop. In any case, as we have seen, progress in AI has been so slow, and is likely to become even slower as more

difficult problems arise, that it will take considerable perseverance to maintain an enthusiasm for such speculations.

In a companion volume, *Whoever Said We Would Colonise Space?*, it was concluded that, unless there is some, at the moment, unforeseeable breakthrough in suspending the time dimension, it will be impossible for carbon-based life forms to travel the huge distances necessary to colonise planets in other solar systems. Increasingly, unmanned probes, rather than human beings, are carrying out the exploration of the solar system and beyond. Inevitably, those space probes will be endowed, as far as it possible, with intelligent capabilities to enable them to carry out autonomous decision-making:

> **The greatest single stimulus to the evolution of mechanical intelligence is the challenge of space.**
>
> *Arthur C. Clarke (2003), Tomorrow's explorers, in Michael Benson (ed.), Visions of the Interplanetary Probes, New York: Harry N. Abrams, Inc.*

If intellectually superior artificial life forms are developed on Earth then, rather than condemn them to a life of frustration on this planet, why do we not entrust to them the mission of dispersing intelligent life throughout the universe?

Name Index

Subject Index