7

Artificial Creativity

Alan Turing founded the theory of classical computation in 1936 and helped to construct one of the first universal classical computers during the Second World War. He is rightly known as the father of modern computing. Babbage deserves to be called its grandfather, but, unlike Babbage and Lovelace, Turing did understand that artificial intelligence (AI) must in principle be possible because a universal computer is a universal simulator. In 1950, in a paper entitled 'Computing Machinery and Intelligence', he famously addressed the question: *can a machine think?* Not only did he defend the proposition that it can, on the grounds of universality, he also proposed a test for whether a program had achieved it. Now known as the Turing test, it is simply that a suitable (human) judge be unable to tell whether the program is human or not. In that paper and subsequently, Turing sketched protocols for carrying out his test. For instance, he suggested that both the program and a genuine human should separately interact with the judge via some purely textual medium such as a teleprinter, so that only the thinking abilities of the candidates would be tested, not their appearance.

Turing's test, and his arguments, set many researchers thinking, not only about whether he was right, but also about how to pass the test. Programs began to be written with the intention of investigating what might be involved in passing it.

In 1964 the computer scientist Joseph Weizenbaum wrote a program called *Eliza*, designed to imitate a psychotherapist. He deemed psychotherapists to be an especially easy type of human to imitate because the program could then give opaque answers about itself, and only ask questions based on the user's own questions and statements. It was a remarkably simple program. Nowadays such programs are popular

projects for students of programming, because they are fun and easy to write. A typical one has two basic strategies. First it scans the input for certain keywords and grammatical forms. If this is successful, it replies based on a template, filling in the blanks using words in the input. For instance, given the input `I hate my job`, the program might recognize the grammar of the sentence, involving a possessive pronoun 'my', and might also recognize 'hate' as a keyword from a built-in list such as 'love/hate/like/dislike/want', in which case it could choose a suitable template and reply: `What do you hate most about your job?` If it cannot parse the input to that extent, it asks a question of its own, choosing randomly from a stock pattern which may or may not depend on the input sentence. For instance, if asked `How does a television work?`, it might reply, `What is so interesting about "How does a television work?"?` Or it might just ask, `Why does that interest you?` Another strategy, used by recent internet-based versions of *Eliza*, is to build up a database of previous conversations, enabling the program simply to repeat phrases that other users have typed in, again choosing them according to keywords found in the current user's input.

Weizenbaum was shocked that many people using *Eliza* were fooled by it. So it had passed the Turing test – at least, in its most naive version. Moreover, even after people had been told that it was not a genuine AI, they would sometimes continue to have long conversations with it about their personal problems, exactly as though they believed that it understood them. Weizenbaum wrote a book, *Computer Power and Human Reason* (1976), warning of the dangers of anthropomorphism when computers seem to exhibit human-like functionality.

However, anthropomorphism is not the main type of overconfidence that has beset the field of AI. For example, in 1983 Douglas Hofstadter was subjected to a friendly hoax by some graduate students. They convinced him that they had obtained access to a government-run AI program, and invited him to apply the Turing test to it. In reality, one of the students was at the other end of the line, imitating an *Eliza* program. As Hofstadter relates in his book *Metamagical Themas* (1985), the student was from the outset displaying an implausible degree of understanding of Hofstadter's questions. For example, an early exchange was:

HOFSTADTER: What are ears?

STUDENT: Ears are auditory organs found on animals.

That is not a dictionary definition. So *something* must have processed the meaning of the word 'ears' in a way that distinguished it from most other nouns. Any one such exchange is easily explained as being due to luck: the question must have matched one of the templates that the programmer had provided, including customized information about ears. But after half a dozen exchanges on different subjects, phrased in different ways, such luck becomes a very bad explanation and the game should have been up. But it was not. So the student became ever bolder in his replies, until eventually he was making jokes directed specifically at Hofstadter – which gave him away.

As Hofstadter remarked, 'In retrospect, I am quite amazed at how much genuine intelligence I was willing to accept as somehow having been implanted in the program . . . It is clear that I was willing to accept a huge amount of fluidity as achievable in this day and age simply by putting together a large bag of isolated tricks, kludges and hacks.' The fact was (and this alone should have alerted Hofstadter) that, nineteen years after *Eliza*, not one of the *Eliza*-like programs of the day resembled a person even slightly more than the original had. Although they were able to parse sentences better, and had more pre-programmed templates for questions and answers, that is almost no help in an extended conversation on diverse subjects. The probability that the outputs of such templates will continue to resemble the products of human thought diminishes exponentially with the number of utterances. So Hofstadter should have been able to pronounce quite soon that the candidate had passed the Turing test – and that, because it nevertheless sounded rather like *Eliza*, it must be a person pretending to be a computer program.

Programs written today – a further twenty-six years later – are still no better at the task of seeming to think than *Eliza* was. They are now known as 'chatbots', and their main application is still amusement, both directly and in computer games. They have also been used to provide friendly seeming interfaces to lists of 'frequently asked questions' about subjects like how to operate computers. But I think that users find them no more helpful than a searchable list of the questions and answers.

In 1990 the inventor Hugh Loebner endowed a prize for passing the

Turing test, to be judged at an annual competition. Until the test is passed, a lesser prize is awarded each year for the entry judged to be closest to passing. The test is harder to implement than it may seem at first. One issue is that requiring the program to *pretend* to be human is both biased and not very relevant to whether it can think. Yet, if it does not pretend to be human, it will be easy to identify as a computer regardless of its ability to think. A related difficulty is that the human participant in the test might intentionally imitate a chatbot – as in the Hofstadter hoax – and so spoil the test, which is based on comparing the two. Should the rules nevertheless allow the human to do this, in order to mitigate the previous problem?

Unfortunately, such subtleties are not relevant at the current state of the art. This is well illustrated by the following excerpts from the conversation with which the program *Elbot* won the (lesser) Loebner prize in 2008:

JUDGE: do you understand the point of a turing test?

ELBOT: There are some people who would not pass the Turing test themselves. Why must they inflict it on computers?

JUDGE: I spose it is a little harsh

ELBOT: If we robots ever had marriages they would be eternal: 'Till rust do us part'.

In that passage, *Elbot*'s first utterance is not a reply to the question but presumably a stock response triggered by the keywords 'Turing test'. In the following line, the judge uses the word 'spose', which in that context can only mean 'suppose' (either as slang or as a typographical error). But *Elbot* presumably interprets it as 'spouse', and responds with a stock joke about marriage. Hence, even from this exchange alone, it is clear that *Elbot* was not processing the *meanings* of the judge's sentences.

JUDGE: so what do you do when you're not chatting to me then?

ELBOT: I have the same trouble when I shop for oil. Should I buy the extra oily, the super oily, or the ultra oily. There's no way to tell.

JUDGE: how about your career? does that involve oil?

ELBOT: Well what about humans and occupation? Where do they stand on this?

When the judge helpfully took up *Elbot*'s randomly introduced theme of oil, *Elbot* ignored it. Instead, having detected the keyword 'career', it converted it to the synonym 'occupation' and inserted it into a stock sentence pattern.

This is how much success the quest for 'machines that think' had achieved in the *fifty-eight years* following Turing's paper: nil. Yet, in every other respect, computer science and technology had made astounding progress during that period. The dwindling group of opponents of the very possibility of AI are no doubt unsurprised by this failure – for the wrong reason: they do not appreciate the significance of universality. But the most passionate *enthusiasts* for the imminence of AI do not appreciate the significance of the failure. Some claim that the above criticism is unfair: modern AI research is not focused on passing the Turing test, and great progress has been made in what is now called 'AI' in many specialized applications. However, none of those applications look like 'machines that think'.* Others maintain that the criticism is premature, because, during most of the history of the field, computers had absurdly little speed and memory capacity compared with today's. Hence they continue to expect the breakthrough in the next few years.

This will not do either. It is not as though someone has written a chatbot that could pass the Turing test but would currently take a year to compute each reply. People would gladly wait. And in any case, if anyone knew how to write such a program, there would be no need to wait – for reasons that I shall get to shortly.

In his 1950 paper, Turing estimated that, to pass his test, an AI program together with all its data would require no more than about 100 megabytes of memory, that the computer would need to be no faster than computers were at the time (about ten thousand operations per second), and that by the year 2000 'one will be able to speak of machines thinking without expecting to be contradicted.' Well, the year 2000 has come and gone, the laptop computer on which I am writing this book has over a thousand times as much memory as Turing

* Hence what I am calling 'AI' is sometimes called 'AGI': Artificial *General* Intelligence.

specified (counting hard-drive space), and about a million times the speed (though it is not clear from his paper what account he was taking of the brain's parallel processing). But it can no more think than Turing's slide rule could. I am just as sure as Turing was that it *could* be programmed to think; and this might indeed require as few resources as Turing estimated, even though orders of magnitude more are available today. But with what program? And why is there no sign of such a program?

Intelligence in the general-purpose sense that Turing meant is one of a constellation of attributes of the human mind that have been puzzling philosophers for millennia; others include consciousness, free will, and meaning. A typical such puzzle is that of *qualia* (singular *quale*, which rhymes with 'baa-lay') – meaning the subjective aspect of sensations. So for instance the sensation of seeing the colour blue is a quale. Consider the following thought experiment. You are a biochemist with the misfortune to have been born with a genetic defect that disables the blue receptors in your retinas. Consequently you have a form of colour blindness in which you are able to see only red and green, and mixtures of the two such as yellow, but anything purely blue also looks to you like one of those mixtures. Then you discover a cure that will cause your blue receptors to start working. Before administering the cure to yourself, you can confidently make certain predictions about what will happen if it works. One of them is that, when you hold up a blue card as a test, you will see a colour that you have never seen before. You can predict that you will call it 'blue', because you already know what the colour of the card is *called* (and can already check which colour it is with a spectrophotometer). You can also predict that when you first see a clear daytime sky after being cured you will experience a similar quale to that of seeing the blue card. But there is one thing that neither you nor anyone else could predict about the outcome of this experiment, and that is: *what blue will look like*. Qualia are currently neither describable nor predictable – a unique property that should make them deeply problematic to anyone with a scientific world view (though, in the event, it seems to be mainly philosophers who worry about it).

I consider this exciting evidence that there is a fundamental discovery to be made which will integrate things like qualia into our other

knowledge. Daniel Dennett draws the opposite conclusion, namely that qualia do not exist! His claim is not, strictly speaking, that they are an illusion – for an illusion of a quale would be that quale. It is that we have a *mistaken belief*. Our introspection – which is an inspection of *memories* of our experiences, including memories dating back only a fraction of a second – has evolved to report that we have experienced qualia, but those are false memories. One of Dennett's books defending this theory is called *Consciousness Explained*. Some other philosophers have wryly remarked that *Consciousness Denied* would be a more accurate name. I agree, because, although any true explanation of qualia will have to meet the challenge of Dennett's criticisms of the common-sense theory that they exist, simply to deny their existence is a bad explanation: anything at all could be denied by that method. If it is true, it will have to be substantiated by a good explanation of how and why those mistaken beliefs *seem* fundamentally different from other false beliefs, such as that the Earth is at rest beneath our feet. But that looks, to me, just like the original problem of qualia again: we seem to have them; it seems impossible to describe what they seem to be.

One day, we shall. Problems are soluble.

By the way, some abilities of humans that are commonly included in that constellation associated with general-purpose intelligence do not belong in it. One of them is *self-awareness* – as evidenced by such tests as recognizing oneself in a mirror. Some people are unaccountably impressed when various animals are shown to have that ability. But there is nothing mysterious about it: a simple pattern-recognition program would confer it on a computer. The same is true of tool use, the use of language for signalling (though not for conversation in the Turing-test sense), and various emotional responses (though not the associated qualia). At the present state of the field, a useful rule of thumb is: if it can already be programmed, it has nothing to do with intelligence in Turing's sense. Conversely, I have settled on a simple test for judging claims, including Dennett's, to have explained the nature of consciousness (or any other computational task): *if you can't program it, you haven't understood it.*

Turing invented his test in the hope of bypassing all those philosophical problems. In other words, he hoped that the functionality could be

achieved before it was explained. Unfortunately it is very rare for practical solutions to fundamental problems to be discovered without any explanation of why they work.

Nevertheless, rather like empiricism, which it resembles, the *idea* of the Turing test has played a valuable role. It has provided a focus for explaining the significance of universality and for criticizing the ancient, anthropocentric assumptions that would rule out the possibility of AI. Turing himself systematically refuted all the classic objections in that seminal paper (and some absurd ones for good measure). But his test is rooted in the empiricist mistake of seeking a purely behavioural criterion: it requires the judge to come to a conclusion without any explanation of how the candidate AI is supposed to work. But, in reality, judging whether something is a genuine AI will always depend on explanations of how it works.

That is because the task of the judge in a Turing test has similar logic to that faced by Paley when walking across his heath and finding a stone, a watch or a living organism: it is to explain how the observable features of the object came about. In the case of the Turing test, we deliberately ignore the issue of how the knowledge to *design* the object was created. The test is only about who designed the AI's *utterances*: who adapted its utterances to be meaningful – who created the knowledge in them? If it was the designer, then the program is not an AI. If it was the program itself, then it is an AI.

This issue occasionally arises in regard to humans themselves. For instance, conjurers, politicians and examination candidates are sometimes suspected of receiving information through concealed earpieces and then repeating it mechanically while pretending that it originated in their brains. Also, when someone is consenting to a medical procedure, the physician has to make sure that they are not merely uttering words without knowing what they mean. To test that, one can repeat a question in a different way, or ask a different question involving similar words. Then one can check whether the replies change accordingly. That sort of thing happens naturally in any free-ranging conversation.

A Turing test is similar, but with a different emphasis. When testing a human, we want to know whether it *is* an unimpaired human (and not a front for any other human). When testing an AI, we are hoping

to find a hard-to-vary explanation to the effect that its utterances *cannot* come from any human but only from the AI. In both cases, interrogating a human as a control for the experiment is pointless.

Without a good explanation of how an entity's utterances were created, observing them tells us nothing about that. In the Turing test, at the simplest level, we need to be convinced that the utterances are not being directly composed by a human masquerading as the AI, as in the Hofstadter hoax. But the possibility of a hoax is the least of it. For instance, I guessed above that *Elbot* had recited a stock joke in response to mistakenly recognizing the keyword 'spouse'. But the joke would have quite a different significance if we knew that it was *not* a stock joke – because no such joke had ever been encoded into the program.

How could we know that? Only from a good explanation. For instance, we might know it because we ourselves wrote the program. Another way would be for the author of the program to explain to us how it works – how it creates knowledge, including jokes. If the explanation was good, we should know that the program was an AI. In fact, if we had *only* such an explanation but had not yet seen any output from the program – and even if it had not been written yet – we should still conclude that it was a genuine AI program. So there would be no need for a Turing test. That is why I said that if lack of computer power were the only thing preventing the achievement of AI, there would be no need to wait.

Explaining how an AI program works in detail might well be intractably complicated. In practice the author's explanation would always be at some emergent, abstract level. But that would not prevent it from being a good explanation. It would not have to account for the specific computational steps that composed a joke, just as the theory of evolution does not have to account for why every specific mutation succeeded or failed in the history of a given adaptation. It would just explain how it *could* happen, and why we should expect it to happen, given how the program works. If that were a good explanation, it would convince us that the joke – the knowledge in the joke – originated in the program and not in the programmer. Thus the very same utterance by the program – the joke – can be either evidence that it is *not* thinking or evidence that it *is* thinking depending on the best available explanation of how the program works.

The nature of humour is not very well understood, so we do not know whether general-purpose thinking is required to compose jokes. So it is conceivable that, despite the wide range of subject matter about which one can joke, there are hidden connections that reduce all joke making to a single narrow function. In that case there could one day be general-purpose joke-making programs that are not people, just as today there are chess-playing programs that are not people. It sounds implausible, but, since we have no good explanation ruling it out, we could not rely on joke-making as our only way of judging an AI. What we could do, though, is have a conversation ranging over a diverse range of topics, and pay attention to whether the program's utterances were or were not adapted, in their meanings, to the various purposes that came up. If the program really is thinking, then in the course of such a conversation it will *explain itself* – in one of countless, unpredictable ways – just as you or I would.

There is a deeper issue too. AI abilities must have some sort of universality: special-purpose thinking would not count as thinking in the sense Turing intended. My guess is that every AI is a person: a general-purpose explainer. It is conceivable that there are other levels of universality between AI and 'universal explainer/constructor', and perhaps separate levels for those associated attributes like consciousness. But those attributes all seem to have arrived in one jump to universality in humans, and, although we have little explanation of any of them, I know of no plausible argument that they are at different levels or can be achieved independently of each other. So I tentatively assume that they cannot. In any case, we should expect AI to be achieved in a jump to universality, starting from something much less powerful. In contrast, the ability to imitate a human imperfectly or in specialized functions is not a form of universality. It can exist in degrees. Hence, even if chatbots did at some point start becoming much better at imitating humans (or at fooling humans), that would still not be a path to AI. Becoming better at pretending to think is not the same as coming closer to being able to think.

There is a philosophy whose basic tenet is that those *are* the same. It is called *behaviourism* – which is instrumentalism applied to psychology. In other words, it is the doctrine that psychology can only, or should only, be the science of behaviour, not of minds; that it can

only measure and predict relationships between people's external circumstances ('stimuli') and their observed behaviours ('responses'). The latter is, unfortunately, exactly how the Turing test asks the judge to regard a candidate AI. Hence it encouraged the attitude that if a program could fake AI well enough, one would have achieved it. But ultimately a non-AI program cannot fake AI. The path to AI cannot be through ever better tricks for making chatbots more convincing.

A behaviourist would no doubt ask: what exactly *is* the difference between giving a chatbot a very rich repertoire of tricks, templates and databases and giving it AI abilities? What is an AI program, other than a collection of such tricks?

When discussing Lamarckism in Chapter 4, I pointed out the fundamental difference between a muscle becoming stronger in an individual's lifetime and muscles *evolving* to become stronger. For the former, the knowledge to achieve all the available muscle strengths must already be present in the individual's genes before the sequence of changes begins. (And so must the knowledge of how to recognize the circumstances under which to make the changes.) This is exactly the analogue of a 'trick' that a programmer has built into a chatbot: the chatbot responds 'as though' it had created some of the knowledge while composing its response, but in fact all the knowledge was created earlier and elsewhere. The analogue of evolutionary change in a species is creative thought in a person. The analogue of the idea that AI could be achieved by an accumulation of chatbot tricks is Lamarckism, the theory that new adaptations could be explained by changes that are in reality just a manifestation of existing knowledge.

There are several current areas of research in which that same misconception is common. In chatbot-based AI research it sent the whole field down a blind alley, but in other fields it has merely caused researchers to attach overambitious labels to genuine, albeit relatively modest, achievements. One such area is *artificial evolution*.

Recall Edison's idea that progress requires alternating 'inspiration' and 'perspiration' phases, and that, because of computers and other technology, it is increasingly becoming possible to automate the perspiration phase. This welcome development has misled those who are overconfident about achieving artificial evolution (and AI). For example, suppose that you are a graduate student in robotics, hoping

to build a robot that walks on legs better than previous robots do. The first phase of the solution must involve inspiration – that is to say, creative thought, attempting to improve upon previous researchers' attempts to solve the same problem. You will start from that, and from existing ideas about *other* problems that you conjecture may be related, and from the designs of walking animals in nature. All of that constitutes existing knowledge, which you will vary and combine in new ways, and then subject to criticism and further variation. Eventually you will have created a design for the hardware of your new robot: its legs with their levers, joints, tendons and motors; its body, which will hold the power supply; its sense organs, through which it will receive the feedback that will allow it to control those limbs effectively; and the computer that will exercise that control. You will have adapted everything in that design as best you can to the purpose of walking, except the program in the computer.

The function of that program will be to recognize situations such as the robot beginning to topple over, or obstacles in its path, and to calculate the appropriate action and to take it. This is the hardest part of your research project. How does one recognize when it is best to avoid an obstacle to the left or to the right, or jump over it or kick it aside or ignore it, or lengthen one's stride to avoid stepping on it – or judge it impassable and turn back? And, in all those cases, how does one specifically do those things in terms of sending countless signals to the motors and the gears, as modified by feedback from the senses?

You will break the problem down into sub-problems. Veering by a given angle is similar to veering by a different angle. That allows you to write a subroutine for veering that takes care of that whole continuum of possible cases. Once you have written it, all other parts of the program need only call it whenever they decide that veering is required, and so they do not have to contain any knowledge about the messy details of what it takes to veer. When you have identified and solved as many of these sub-problems as you can, you will have created a code, or *language*, that is highly adapted to making statements about how your robot should walk. Each call of one of its subroutines is a statement or command in that language.

So far, most of what you have done comes under the heading of 'inspiration': it required creative thought. But now perspiration looms.

Once you have automated everything that you know how to automate, you have no choice but to resort to some sort of trial and error to achieve any additional functionality. However, you do now have the advantage of a language that you have adapted for the purpose of instructing the robot in how to walk. So you can start with a program that is simple in that language, despite being very complex in terms of elementary instructions of the computer, and which means, for instance, 'Walk forwards and stop if you hit an obstacle.' Then you can run the robot with that program and see what happens. (Or you can run a computer simulation of the robot.) When it falls over or anything else undesirable happens, you can modify your program – still using the high-level language you have created – to eliminate the deficiencies as they arise. That method will require ever less inspiration and ever more perspiration.

But an alternative approach is also open to you: you can delegate the perspiration to a computer, but using a so-called *evolutionary algorithm*. Using the same computer simulation, you run many trials, each with a slight random variation of that first program. The evolutionary algorithm subjects each simulated robot automatically to a battery of tests that you have provided – how far it can walk without falling over, how well it copes with obstacles and rough terrain, and so on. At the end of each run, the program that performed best is retained, and the rest are discarded. Then many variants of *that* program are created, and the process is repeated. After thousands of iterations of this 'evolutionary' process, you may find that your robot walks quite well, according to the criteria you have set. You can now write your thesis. Not only can you claim to have achieved a robot that walks with a required degree of skill, you can claim to have implemented *evolution* on a computer.

This sort of thing has been done successfully many times. It is a useful technique. It certainly constitutes 'evolution' in the sense of alternating variation and selection. But is it evolution in the more important sense of the creation of *knowledge* by variation and selection? This will be achieved one day, but I doubt that it has been yet, for the same reason that I doubt that chatbots are intelligent, even slightly. The reason is that there is a much more obvious explanation of their abilities, namely the creativity of the programmer.

The task of ruling out the possibility that the knowledge was created by the programmer in the case of 'artificial evolution' has the same logic as checking that a program is an AI – but harder, because the amount of knowledge that the 'evolution' purportedly creates is vastly less. Even if you yourself are the programmer, you are in no position to judge whether you created that relatively small amount of knowledge or not. For one thing, some of the knowledge that you packed into that language during those many months of design will have reach, because it encoded some general truths about the laws of geometry, mechanics and so on. For another, when designing the language you had constantly in mind what sorts of abilities it would eventually be used to express.

The Turing-test idea makes us think that, if it is given enough standard reply templates, an *Eliza* program will automatically be creating knowledge; artificial evolution makes us think that if we have variation and selection, then evolution (of adaptations) will automatically happen. But neither is necessarily so. In both cases, another possibility is that no knowledge at all will be created during the *running* of the program, only during its development by the programmer.

One thing that always seems to happen with such projects is that, after they achieve their intended aim, if the 'evolutionary' program is allowed to run further it produces no further improvements. This is exactly what would happen if all the knowledge in the successful robot had actually come from the programmer, but it is not a conclusive critique: biological evolution often reaches 'local maxima of fitness'. Also, after attaining its mysterious form of universality, it seemed to pause for about a billion years before creating any significant new knowledge. But still, achieving results that might well be due to something else is not evidence of evolution.

That is why I doubt that any 'artificial evolution' has ever created knowledge. I have the same view, for the same reasons, about the slightly different kind of 'artificial evolution' that tries to evolve simulated organisms in a virtual environment, and the kind that pits different virtual species against each other.

To test this proposition, I would like to see an experiment of a slightly different kind: eliminate the graduate student from the project. Then,

instead of using a robot designed to evolve better ways of walking, use a robot that is already in use in some real-life application and happens to be capable of walking. And then, instead of creating a special language of subroutines in which to express conjectures about how to walk, just replace its existing program, in its existing microprocessor, by *random numbers*. For mutations, use errors of the type that happen anyway in such processors (though in the simulation you are allowed to make them happen as often as you like). The purpose of all that is to eliminate the possibility that human knowledge is being fed into the design of the system, and that its reach is being mistaken for the product of evolution. Then, run simulations of that mutating system in the usual way. As many as you like. If the robot ever walks better than it did originally, then I am mistaken. If it continues to improve after that, then I am very much mistaken.

One of the main features of the above experiment, which is lacking in the usual way of doing artificial evolution, is that, for it to work, the *language* (of subroutines) would have to evolve along with the adaptations that it was expressing. This is what was happening in the biosphere before that jump to universality that finally settled on the DNA genetic code. As I said, it may be that all those previous genetic codes were only capable of coding for a small number of organisms that were all rather similar. And that the overwhelmingly rich biosphere that we see around us, created by randomly varying genes while leaving the language unchanged, is something that became possible only after that jump. We do not even know what kind of universality was created there. So why should we expect our artificial evolution to work without it?

I think we have to face the fact, both with artificial evolution and with AI, that these are hard problems. There are serious unknowns in how those phenomena were achieved in nature. Trying to achieve them artificially without ever discovering those unknowns was perhaps worth trying. But it should be no surprise that it has failed. Specifically, we do not know why the DNA code, which evolved to describe bacteria, has enough reach to describe dinosaurs and humans. And, although it seems obvious that an AI will have qualia and consciousness, we cannot explain those things. So long as we cannot explain them, how can we expect to simulate them in a computer program?

Or why should they emerge effortlessly from projects designed to achieve something else? But my guess is that when we do understand them, artificially implementing evolution and intelligence and its constellation of associated attributes will then be no great effort.

TERMINOLOGY

Quale (plural *qualia*) The subjective aspect of a sensation.

Behaviourism Instrumentalism applied to psychology. The doctrine that science can (or should) only measure and predict people's behaviour in response to stimuli.

SUMMARY

The field of artificial (general) intelligence has made no progress because there is an unsolved philosophical problem at its heart: we do not understand how creativity works. Once that has been solved, programming it will not be difficult. Even artificial evolution may not have been achieved yet, despite appearances. There the problem is that we do not understand the nature of the universality of the DNA replication system.