Contents

AI lecture 8

Jocelyn Ireson-Paine

February 14, 1996

1 Introduction — criticisms of AI

There are many different levels at which one can criticise AI. These vary from details of a particular method or model, up to (for example) grand criticisms of the kind that Searle, Dreyfus and Penrose make.

I'll start with a list of criticisms at different levels, to demonstrate the variety involved. Then I'll discuss Searle, Dreyfus and Penrose in more detail. I shall put a big bibliography of philosophical articles, by Chalmers, into the AI box at some stage, with lots of references to such matters.

2 Goals and planning — criticisms

In this section, I'll focus on the notion of *goal*, and I'll trace a trajectory through "criticism space", ending up with the complete abolition of goals and even computation.

In the early 60s, Newell Simon and Shaw introduced the General Problem Solver, GPS. This marked the start of the *planning* paradigm. I discuss GPS in my second set of tutorial notes. For now, all you need to know is that GPS sees problems in terms of *goals* to be achieved. Each goal describes what state the world must be in for the problem to have been accomplished. GPS also has a table of *operators*. These are the basic actions available to the problem solver: the table specified how each action, if carried out alone, transforms the state of the world. Using a *means-ends* algorithm, GPS then attempts to find a sequence of actions which transforms the initial state of the world into one of the goal states.

For references to GPS, Boden gives a clear non-technical account in pages 354–357 of *AI and Natural Man*. Winston has a more detailed account, with examples of how one formalises problems. One of the original articles appears in *GPS, A Program that Simulates Human Thought*, by A Newell and H A Simon, in *Computers and Thought* edited by E A Feigenbaum and J Feldman (RSL: per 19668 d 335; PSY: KH:F 032). In the latter article, the authors state that they believe the subjects they tested were manipulating symbols in the same way as GPS (and hence presumably had an explicit representation for their goals). This article also appears in *Readings in Planning* (page 59), location below.

2.1 Criticisms

- By incorporating theorem-proving methods into a means-ends planner, one can build a planner which allows a wider variety of problem worlds than GPS. But the underlying planning method is the same. *STRIPS: A New Approach to the Applications of Theorem Proving to Problem Solving* by Fikes and Nilsson. Published in 1971, reprinted in *Readings in Planning* (page 88) edited by Allen, Hendler and Tate (PSY KH:A 427). Commentary on page 57.

- General-purpose planners like STRIPS and GPS use general-purpose search heuristics. It is inevitable that planners using such heuristics will, when solving complex problems, get caught in a combinatorial explosion. To overcome this, we can make the planner distinguish between details and essential information. When building our initial plan, we ignore the details. After we've made the plan, we then refine it by gradually introducing the details and reconstructing those portions of the plan that need them. This gives us ABSTRIPS. *Planning in a Hierarchy of Abstraction Spaces* by Sacerdoti. Published in 1974, reprinted in *Readings in Planning* (page 98). Commentary on page 57.

- The planners above distinguish between planning and execution — it's assumed that all changes in the environment can be known to the planner. But this assumption is unrealistic, and so we must be able to repair our plans as they're being executed, if something unexpected happens. *Learning and Executing Generalized Robot Plans* by Fikes, Hart and Nilsson. Published in 1972, reprinted in *Readings in Planning* (page 189). Page 187 is a general commentary on the problem of unexpected events.

- The notion of goal is too crude, being a binary partition of the environment into states that precisely achieve some outcome versus states that don't. This is unrealistic. Instead of allocating complete desirability to the goal states, and complete undesirability to all the others, each state will have a particular *utility*. For instance, in filling a petrol tank, you want to spill as little petrol as possible; but it may not be possible to fill it without spilling any. So we have a set of states whose utility depends on the amount of spilt petrol.

  Moreover, realistic agents will usually have more than one desirable outcome in mind. STRIPS-style goal-based planning is no help in building an agent which can trade-off priorities between different desirable outcomes.

  To overcome these two defects, we need to find a new semantics for goals, in terms of continuously variable preferences between outcomes. This should be based on decision theory. See pages 210–212 of *Planning and Control* by Dean and Wellman (Morgan Kaufmann 1991; PSY KH:D 034). This account is based on a paper published in 1991, *Preferential semantics for goals* by Wellman and Doyle, in *Proceedings AAAI-91*.

- So far, we've assumed that agents work with *world models*. These are symbolic descriptions of the world, in which different tokens uniquely represent distinct individuals. See for example the POETIC system which I referred to in my first lecture. One of the things you do when planning is to use this model to simulate the way in which your actions will change the world, bringing you nearer to or further from a desirable goal.

  But the type of behaviour generated by a central controller (planner plus execution mechanism) acting on a world model is inflexible, "brittle" and slow. We should investigate an alternative approach where behaviour emerges naturally as an effect of co-operation between many simple modules. Such an agent will still have goals, but they're represented in a different way from goals in conventional planners. In particular, they use a *deictic* representation. Instead of using separate tokens to identify different objects, the agent has a self-centered representation, and uses symbols like `the sprayer I am holding now`, or `the food I can see in front of me`. This means less search during planning, and less perceptual decoding. *Situated Agents Can Have Goals* by Pattie Maes. Published in 1990. From *Designing Autonomous Agents* edited by Maes (PSY KH:M 026).

- The idea that robots should explicitly represent their goals is wrong. It's based on a naive *folk-psychological* view. Just as astrology is a naive pre-scientific theory of the night sky, so folk-psychological entities like goals are pre-scientific constructs. They arise from our own introspection; but there's no reason to expect our introspective perceptions of our own minds to reflect reality any more closely than did the Babylonian's perceptions of the night sky. This stance is called *eliminative materialism*.

  Not only is it bad cognitive science to try and find such entities in the mind, it's bad engineering to build them into our robots. Doing so will lead to inefficient, incapable, systems. *Taking Eliminative Materialism Seriously: A methodology for Autonomous Systems Research* by Tim Smithers, from *Towards a Practice of Autonomous Systems*, edited by Varela and Bourgine (MIT 1992: PSY KH:V 042; RSL M92.C00938).

- In any case, the notion of representation is too weak! Instead of explaining cognition as the manipulation of representations by computational systems, we should see it as state-space evolution in dynamical systems. *Explaining the Behaviour of Springs, Pendulums, and Cognizers* by Michael Wheeler. Sussex University CogSci report CSRP 284. Published June 1993. AI Box photocopy W87.

- But unless these dynamical systems embody certain non-local quantum effects, they may not be able to implement the "holistic perception" required by human cognition. No Turing-equivalent computer can do so (and, as far as I can see, none of the other dynamical systems we can currently build could either). Penrose, see below.

3 Grander criticisms — Penrose, Searle and Dreyfus

I'll now talk about three "ingredient X" arguments, each of which claims that the minds can do something that a Turing-equivalent computer can't. These all take the form "Humans can do A; but computers can't, because they lack ingredient X".

Such arguments are often inspired by comparing the the inflexibility of conventional AI to human creativity, intuition or adaptability. Some claim that connectionism may be able to go further than classical symbolic AI. Others don't.

Summary:

- *Searle*. John Searle, professor of philosophy at Berkeley. Known in AI for the *Chinese Room* argument, popularised (I think) in a mid-80's Reith lectures . A = attain awareness; X = the right stuff.

- *Dreyfus*. Hubert Dreyfus, professor of philosophy at Berkeley. Known for a long history of criticisms against AI, starting in 1961. Known for his books *What Computers Can't Do* and (with his brother) *Mind over Machine*. A = attain human-level skill; X = holistic processing.

- *Penrose*. Roger Penrose, professor of mathematics at Oxford, working on quantum physics and cosmology. Known in AI for his book *The Emperor's New Mind*. A = consciousness and behaviour; X = the right stuff (particular quantum phenomena).

4 Searle

For his original paper, including the Chinese Room argument, see *Minds, Brains and Programs*, in *Behavioural and Brain Science* volume 3 1980. Reprinted in *The Mind's I* edited by Hofstadter and Dennett (PSY AA:H 067; Philosophy; Hooke). Also reprinted in *Mind Design* edited by Haugeland (PSY KH:H029).

The B and BS paper is followed by twenty-eight replies from Searle's critics, these are also worth reading. They have been omitted from *The Mind's Eye* and *Mind Design*, but the editors discuss a few in the *Reflections* section following the *Mind's Eye* reprint. Searle himself discusses a few of the possible replies in his original paper; apparently he put a first draft to several AI people before publishing the final version. There's a summary of his argument on pages 269–271 of Crevier (PSY KH:C 086).

Main argument: Assume, says Searle, that some A.I. researchers have devised a program that speaks Chinese. You ask it questions (in Chinese), and it replies with appropriate answers (also in Chinese), drawing information where necessary from some memory of imaginary events that its programmers have implanted. So if it had been given memories of dining in a Chinese restaurant, you might ask "What did you think of last night's meal?", and it might reply "The crispy duck was fine, but I'm not very good with chopsticks, so I didn't get as much as I wanted."

Now, this program is essentially a set of symbol-manipulating rules encoded in Prolog, Pascal, or whatever. Take these rules, and write them down on paper. Put them in a small lockable windowless room, together with a supply of paper, filing cabinets, pencils, and rubbers. To one outside wall of this room, affix a TV camera and scanner. This camera can be shown Chinese characters. It digitises them, and sends the results along a cable to a VDU inside the room. To another outside wall, affix a ticker-tape machine. It can print in Chinese, and is controlled by commands coming along another cable from a keyboard inside the room. Finally, find a person to do the clerical work — a graduate accountancy reject perhaps. This clerk is to be locked in the room; his job is to copy numbers off the internal screen, process them strictly according to the rules you've provided, and type out the results on the keyboard. They'll be sent to the ticker-tape machine, and printed out as Chinese.

You now have a person playing the part of your Chinese-speaking program. To an outsider, the combination of room, ticker-tape printer, and TV camera, behaves just like a native Chinese. But — a native Chinese feels that he *understands* the questions he's asked. The clerk feels no such thing. All he's doing is manipulating symbols that to him are meaningless. You may ask a question about woks and bean sprouts, or Beijing and student riots, but he doesn't *know* it's about these — all he knows is a string of meaningless symbols to be turned into other equally meaningless symbols. He is an example of a system that's working by formal symbol-manipulation; he doesn't understand the meaning behind them; hence no other formal ssymbol-manipulation system can either.

One reply is the *systems reply* which is functionalist in nature. What matters is the formal identity between the room and the brain. Understanding resides in the entirety of the system (the room plus the rules plus all the pencils and paper), not in one little part of it, the man. Searle tries to disguise this by making the rest of the system appear much smaller than it is; in reality, the "room", with its filing cabinets and rulebooks would cover the whole of the Earth's surface, and the man would play as insignificant a role as does one neuron in a brain. (See also the Churchland's reply: Crevier page 270.)

Searle's attitude to functionalism: formal identity of structure is not sufficient. The brain (or other thinking system) must also have the "right causal powers". I'm not clear on exactly what he means by this, but it seems to require that they be made out of the right kind of material. In the same way, a simulation of photosynthesis or of an ice-cream factory may have the right formal structure, but it still won't emit oxygen, or ice creams.

5 Dreyfus

Dreyfus claims that following rules, as is done by expert systems is not enough. It can't capture the richness, the "thickness" of human experience. See *Mind over Machine* by Dreyfus and Dreyfus (PSY KH:D 081). For a summary, see Crevier pages 127–132.

Part of the Dreyfus' argument is based on their five-stage model of progress from novice to expert (page 22 of *Mind over Machine*)

1. Novice. Operates by consciously-learnt context-free rules. Lacks any sense of the overall task.

2. Advanced beginner. Uses more sphisticated rules, which refer to situational elements as well as context-free ones. These situational elements are features such as the pattern of behviour which distinguishes a drunken from a sober driver. They're learnt by experience, and the advanced beginner can't formalise them.

3. Competent. Has now learnt to recognise many many context-free and situational elements. Still lacks any sense of their overall importance to the task, and rapidly becomes overwhelmed. Tries to overcome this by hierarchical goal-based planning. This hierarchical decomposition of the task means that, at any time, the competent pays attention only to that small number of features relevant to a particular subgoal, thus avoiding being overwhelmed.

4. Proficient. Most of the time, now performs his task intuitively, without analytical thought. But this deep involvement in the task will be broken when certain elements present themselves as particularly important. The proficient then stops and thinks analytically about what to do next.

5. Expert. Performs his task intuitively, almost all the time. Occasionally has to stop and deliberate, but this involves critical reflection on his intuitions, rather than goal-based planning.

So this is a progression from rule-based problem-solving to a different approach based on matching against past experiences. The first is similar to classical symbolic AI, which divides problems down into bits, divides the job between different components, and puts the results together. The second involves some kind of holistic pattern recognition.

Classical AI assumes the mind also divides problems down into bits, divides the job between different components, and puts the results together. Hence it's not a good model for expert performance. Digital computers require one to organise tasks in this way: hence they can't achieve expert performance. In Dreyfus' terms, such systems he calls "machines".

What is required for holistic pattern recognition is something like a holographic recogniser (page 60), example in *Transforming the prospects for robot vision*, AI photocopy B165. Like computers, such systems are physical systems. Unlike computers, they don't divide the job up into separate pieces, giving one to each component. Dreyfus does not call systems of this kind machines.

Dreyfus does not claim machines (in our sense) can't behave intelligently; just that digital computers and other machines (in his sense) can't. He does not deny the brain is a machine (in our sense).

6 Penrose

Penrose believes that the brain may use certain quantum phenomena which enable it to solve non-Turing-computable problems. These include mathematical intuition and musical creativity, because they require a kind of holistic global perception that no algorithm can provide. He describes these views in detail in *The Emperor's New Mind* (KH:P something).

There is a demonstration that the brain can compute what's not Turing-computable (I think) on page 64 of his book.

6.1 Matter that can compute the uncomputable

So what does it mean to say that we can build a machine to solve non-Turing-computable problems? David Deutsch, who is a quantum physicist working at the Maths Institute in Oxford, has argued that computability lecture) is not a purely mathematical notion, but is an abstraction from physics. He says that the Turing machine is an abstract description of those basic operations that, because of the nature of our universe, we find easy to perform. In other universes, with other laws of physics, what's computable might be quite different.

According to Deutsch, many mathematicians and computer scientists regard computability as a mathematical property of numbers, functions, etc in the same way that (say) oddness and evenness are. But it isn't. It's just an abstraction from our physical experience. It is logically possible that there could be physical systems which could perform computations beyond the power of any Turing machine (and hence, of any digital computer). I'll use the phrase *Turing-equivalent computer* again to denote anything that has the same computational power as a Turing machine, i.e. any digital computer.

For example, let's suppose we have some scheme for arrangine all the possible computer programs in some language such that each is indexed by a unique integer. So `begin end` gets 1, `begin stop end` gets 2, `begin a:=1 end` gets 3, and so on. Now imagine that we have found a strange crystal which is perfectly opaque to some wavelengths of electromagnetic radiation, perfectly transparent to others, and explodes with yet others. On further investigation, it turns out that is opaque to exactly those wavelengths of radiation whose length in metres is the index of a program that will halt. It's transparent to exactly those wavelengths of radiation whose length in metres is the index of a program that will never halt. If illuminated with any other wavelength, it explodes.

In our universe, this is (as far as we know!) impossible. But it's an impossibility dictated by the laws of physics, not those of logic. In other universes, perhaps such a "halting crystal" could exist.

But is it possible for matter that goes beyond computability to exist in *our* universe? If it is, then that's important to AI. The brain could be made of such matter, and hence have computational powers that go beyond any Turing-equivalent computer.

Penrose believes such matter can exist, and that the brain is composed of it. Deutsch believes such matter can exist, but that the brain is not composed of it. However, says Deutsch, we may be able to build *computers* composed of it. So in Deutsch's view, the computer might eventually outdo the brain. In Penrose's view, the brain outdoes the computer. See *Emperor's New Mind* page 146.

Let me now describe each case.

6.2 Deutsch — computers may use matter that outdoes the brain

I shall now try to describe Deutsch's scheme for building a *quantum computer*. He does not actually try to use some wierd new form of matter like my hypothetical halting crystal. Instead, he exploits quantum phenomena that exist everywhere, but which are usually invisible at a macroscopic scale.

Before I can do so, I must delve into quantum physics. Most of Deutsch's papers on this are very technical, and you're only likely to be able to read them if you start working on such things professionally (but if you do, it will provide — albeit not for ten or fifty years, perhaps — a major advance in computation.) There's a comparatively non-technical account in *Mind, Brain and the Quantum* by Lockwood (PSY AA:L 081), chapter 14.

Summary of what you must know in order to build a quantum computer.

- Consider an electron or a nucleus in a magnetic field. You can imagine it as a ball spinning about its axis of rotation. Electrons and nuclei have an electric charge. Because they're spinning, this causes a magnetic field which points along the axis.

  The axis of rotation can point in various directions. But — **first wierdness** — unlike in everyday experience, the directions are *quantised*. Only a finite number of directions is allowed. This number depends on the particle's electric charge.

  The simplest possibility is that the particle could be pointing "up", or "down", but nowhere in between. In this case, we say that the particle is either "spin up" or "spin down". I will ignore particles with more than two possible spin directions in the rest of these notes.

- As I said in the 6th week lecture, today's computers represent numbers in *binary*. Each number is represented as a series of binary digits or *bits*. For example, the number 3 would be represented as binary 11, and the number 8 as binary 1000. We can encode instructions in binary too. Each bit is stored in a separate memory cell.

- In conventional computers, these cells are built from cross-coupled transistors. But if we can find some way to make electrons or nuclei hold still, we could use a single particle to represent a bit. Use an electron with spin up to represent a 0; an electron with spin down to represent a 1. It would be possible to read out the state of a bit, or to change it, using a method related to nuclear magnetic resonance. We could then build an entire computer memory out of cells, each of which holds a spin-up or spin-down particle.

- In everyday physics, any object can exist in only one state at a time. You can't have a door that's both open and closed, or a ball that's spinning both clockwise and anticlockwise. But in quantum physics — **second wierdness** — you can. Particles can exist in a *superposition of states*, also called a *mixed state*. So you can have an electron that's both spin-up and spin-down at the same time.

- Actually, each component state has a probability associated with it. So our mixed electron is not just spin-up+spin-down, but is (say) 40% spin-up + 60% spin-down, or perhaps 27% spin-up + 73% spin-down. You can also have *pure states*: it's possible, even in quantum physics, for our electron to be 100% spin-up + 0% spin-down.

- This means that we could prepare our computer memory so that some or all of the cells are in mixed states. Think of such a memory as a collection of film strips lying on top of one another. Each strip is a memory in some particular collection of pure states. Each frame in the strip is one cell, one particle. The entire set of superimposed strips is a memory in a collection of mixed states.

- It's possible to build a CPU which, given a mixed state, processes it as though it were processing each of the component states independently. This gives us a kind of parallel processing. We put in a collection of memory cells in mixed states, and we get out a collection of cells in different mixed states. Our CPU is processing all the film strips at the same time.
- This gives us a form of parallel processing for free. If we'd built a conventional (non-quantum) parallel computer, we'd need one CPU for each strip. But here, we have one CPU which is, in effect, processing all the strips simultaneously.
- So far, what I've described is something which carries out a lot of computations in parallel. Since each of these can be done on a serial computer, the quantum machine can't do anything a Turing-equivalent computer can't do. But it can do so *faster*. So it is more powerful in a weak sense; it pushes back the limitations of complexity theory I described last week. [1]

[2]

6.3 Penrose — the brain uses matter that outdoes computers

As I said above, Deutsch believes that we can make computers which use quantum effects to outdo conventional computers — to compute Turing-computable things faster than any Turing machine could. But he does not believe the brain does so. So in Deutsch's view, the brain is no more powerful than a Turing-equivalent computer.

Penrose, on the contrary, believes it is. He also believes that we could build a Deutsch-style quantum computer — *Emperor's New Mind p 401*. But he doesn't believe that its kind of quantum effects would be any use to the brain — *Emperor's New Mind* p 402.

What effects does he want to use? In last week's lecture, I said that one of the problems no Turing-equivalent computer can solve is this: Consider a set of polygons. Is there an algorithm that will tell you whether how to given any set of polygons, it's possible to cover an infinite plane with no gaps and no overlaps? Obviously one can do this for some sets, e.g. that containing just one square. But it can't be done in general.

Going on from that, it may be that for some particularly wierd sets of polygons, there are no rules that tell you how to cover the whole plane with them. I.e. this problem would also be non-Turing-computable. Penrose speculates that may be the case for the *Penrose tiles* shown on pages 136–137 of *Emperor's New Mind*. He makes this speculation on page 438 (bottom paragraph). Apparently, when laying

---

[1] Actually, there's still a snag. We can't read out these states directly. When you *observe* a mixed state, it *collapses* into one of the component states, with appropriate probability — **third wierdness**. So suppose you prepare one million electrons, each 40% spin-up + 60% spin-down. Now you measure the spin of each. You will find that 40% of the electrons (on average) come out as spin up, and 60% as spin down. But it's entirely unpredictable which come out as which.

So when we read out the result of our computation, we only get the result of processing one of these strips. So although our computer has done the parallel computations for free, we've lost all but one of the results. So it might appear that we're still no better off than with a normal computer. However, Deutsch has been able to show that's not true. His kind of quantum computer can still get some results faster — see pages 248–249 of Lockwood.

[2] Michael Stannett has argued that one can go beyond Turing-computability in a different way. See his description of the X machine in *X-Machines and the Halting problem: Building a Super-Turing Machine*. It's available as AI photocopy S171. The conclusion discusses the relevance of this to Deutsch and Penrose.

down the tiles, you may have to look arbitrarily far ahead to check there are no overlaps or gaps.

Suppose now that we could find a crystal which was built up from unit cells with the same shape as Penrose tiles. *If* we believe no Turing-equivalent computer could work out how to cover the plane with Penrose tiles, then, just by being able to crystallise itself, this crystal is doing something which is not Turing-computable. Perhaps it uses some kind of holistic non-local quantum effect to do so (page 437).

In which case, perhaps the brain uses the same kind of non-local quantum effect when it does intuitive, judgemental thought (see pages 411–413 for examples).

There are a lot of assumptions here!

7 The Turing Test

I should talk about this, since it often comes up in AI. Essentially, it is an operational definition of intelligence, which Turing put to answer the question "can a machine think"? He introduced it in *Mind*, volume 54, Number 236 (1950). It's reprinted (abridged) in *The Mind's I* edited by Dennett and Hofstadter, (PSY AA:H 067) and in *Computers and Thought* edited by E A Feigenbaum and J Feldman (PSY: KH:F 032).

Do not confuse it with Turing machines. The two are separated by fifteen years, and have entirely different objectives.

Turing tried to answer the question "Can machines think?". He began by defining his terms. To judge the quality of something's thought, he said, we should be concerned with the symbols it emits; not with irrelevant properties like its physical appearence, strength, or chemical makeup. Our criterion will be how well these symbols compare with those that a human would emit in similar circumstances. As symbols are our concern, we should consider the most versatile symbol-manipulating machine we know: the digital computer.

So the question reduces to "can we build a computer that is conversationally indistinguishable from a human?".

To answer this, Turing invented the "imitation game". As often described, you have three rooms. There is an interrogator in room 1, a computer in either 2 or 3, and a human in the other room. Room 1 contains a teletype linked to the computer, and also to another teletype in the room where the human is. The interrogator does not know who is in 2 and who is in 3, only that there is a computer in one of them and a human in the other. He (or she) can type anything he likes into the link, to either room; and of course, he'll get a typed reply back. If, after "conversing" in any way he likes for as long as he likes, he can't tell which is the computer and which the human, the computer has passed the Turing Test.

The type of test Turing actually described is slightly more complex (see, e.g., Hofstadter). But it reduces to the same idea: a behavioural test for conversational indistinguishability from a person. Note again that the interrogator is allowed to type about anything at all (and in any language he wants). Not just arithmetic or chess, but poetry, politics, philosophy …

Why is this relevant to AI? In Penrose's view, we could distinguish an intelligent Turing-equivalent computer from a human by posing it a question which requires non-algorithmic methods (page 64).

In Dreyfus' view, we could distinguish an expert human from an intelligent computer (though not from an intelligent connectionist system) by looking at its expertise.

In Searle's view, we could not build a computer that's indistinguishable from a human. But the computer would have no subjective awareness. This is an odd view. See *An Unfortunate Dualist* by Smullyan in *Mind's Eye*. [3]

[3] French has argued that the Turing test is not ideal: it probes subcognitive phenomena that are irrelevant to its aim. See *Subgognitive Probing: Hard Questions for the Turing Test* by Robert French, AI box copy F46. From *Proc. 10th Ann. Conf. Cog. Sci. Soc*, Montreal 1988.