Contents

AI lecture 1

Jocelyn Ireson-Paine

February 14, 1996

1 Introduction

This is the first set of notes for my eight lectures on the fundamentals of AI.

In this first lecture, I'll discuss what AI is, give some examples and a bit of history, distinguish symbolic AI from connectionist AI, and then give two examples of symbolic-AI models of memory.

2 What is AI?

The popular view is that it's the quest to build an intelligent machine. This raises a number of questions.

- Can a machine be intelligent?
- What do we mean by intelligence and intelligent behaviour anyway? How does it depend on one's environment?
- Can a machine have self-awareness? If not, what property of the mind makes it possible for us to do so?
- Could a machine act intelligently but not be self-aware? If so, what's the point of consciousness?
- Can one build a machine that learns from its experience, or would its real-world knowledge have to be programmed in in advance? If the latter, how do we go about formalising our knowledge of the world in a suitable notation?
- Would a machine think logically? Would it suffer from optical illusions, or make mistakes in reasoning?
- Would it have emotions? If so, are they a necessary component of intelligence? If not, why do we?
- Would such a machine be a computer? Is computation actually the basis of the mind, or of intelligence in general? Perhaps some other discipline, such as dynamical systems theory, is needed instead. See also my final lecture.

These shade into questions about the nature of intelligence in general. Are there some characteristics that any intelligence, whether man, machine, or Martian, must have? In an essay on communication with extraterrestrial intelligence, Marvin Minsky in *Why intelligent aliens will be intelligible* (photocopy M131 in the library), has argued that there are - that any intelligence will divide the world into things versus actions and events, will have a notion of plan and goal, and will even have some idea of economics! Minsky believes the latter to be necessary whenever there is a limit on one's resources, which is always.

So which properties of our minds are universal? Surprisingly to those who see computers as coldly logical machines, the fact that we make mistakes, including mistakes in perceptual interpretation, is almost certainly one; and any intelligent machine will also do so. In the early days of computing, it seemed that one might be able to solve many problems just by blind search, trying out all possible solutions. In fact, the time taken to solve almost any interesting problem by this method grows exponentially with the size of the problem — the so-called combinatorial explosion, which I refer to when discussing chess. Work in game-playing, translation, and planning, amongst other fields, soon demonstrated that blind search isn't feasible except for very tiny problems. Instead, computer programs, like us, need to rely on cleverly coded real-world knowledge, on special tricks, on heuristics or rules of thumb that work most of the time but not always. In perceptual interpretation, it's often necessary to hypothesise on the basis of incomplete data. [1]

3 What is AI (2)?

The only true definition is "that which AI people do". Looking at the early history of AI, there were already a number of different approaches by the end of the 1950s. For some examples and commentary, see Chapter 8 of *The Guide to Expert Systems*, by Alex Goodall (Learned Information, 1985; RSL Comp BD 36; PSY KH:G 061).[2] Note:

- The cybernetic approach. Derived from cybernetics, the study of control and self-organising systems. The idea was to put together a very large number of simple and similar elements, and study the way in which (hopefully) the system "learnt" to respond to perturbations in its environment by acting so as to negate their effect. Intelligent behaviour was a by-product of this internal goal-seeking or homeostasis. See Ross Ashby's homeostat.

- The neural-net approach. The elements are simulated neurons. Then usually simulated in hardware; now, in software. Many different levels of biological fidelity are possible. See Grey Walter's turtles, and the Perceptron.

- The symbolic or programming approach. One writes a symbol-manipulating program for a digital computer. Exemplified by the Advice Taker.

So right at the start, there were different approaches and methods. This has persisted. AI has few unifying principles and lots of disagreements about its objectives and methods. Too many practitioners spend the time mud-slinging, rather than trying to understand different approaches and see what they have in common.

Like psychology, AI is too large for any one person to claim that their work spans it all. Most researchers work on well-defined subtopics, and some do not see themselves as having much connection with the rest of the subject. For example, Steve Muggleton in the Programming Research Group works on "Inductive Logic Programming" — a form of machine learning where a computer program tries to learn general logical rules from specific instances. Although any complete artificial intelligence would need to be able to learn, Steve never claims to be working on AI. [3]

[1] Note that one doesn't always need to carry out a complete interpretation anyway. When crossing the road, it's usually sufficient to judge the contact time of an approaching car from the apparent enlargement of its image. There's no need to spend time working out its make or year of manufacture. In other situations, more may be necessary — when cycling, it may be useful to know which makes are driven by aggressive young men, so you can avoid them.

[2] See also my tutorial notes

[3] For examples of ILP, see *Predicting secondary structure of proteins with logic-based machine learning* by Muggleton, King and Sternberg (AI photocopy M166). Unlike many connectionist learning systems,

Like ILP, neural networks and genetic algorithms are also kinds of machine learning, but it would be rare to find an individual whose work covered more than one of these three topics. In fact, there are many different designs of neural net and genetic algorithm, and most researchers will specialise in just one kind.

4 A definition of AI

Here's a definition which I think most would agree on, from *Computers and Thought* page xx.

a The empirical study of existing intelligent systems. As shown later, This includes animals as well as people, and activities such as locomotion and low-level vision as well as game-playing and other examples of "rational" thought.

b The theoretical study of possible intelligences, and the general principles underlying them.

c Using the above approaches to solve practical problems, i.e.

  c.1 Trying to deal with the problems of existing intelligences — for example, learning problems and emotional disorders.

  c.2 Designing useful intelligent (or semi-intelligent) machines.

The definition above is a clear description of AI's objectives, but it's harder to say what AI people should be doing in order to reach them. Different parts of the subject seem to be mutually incompatible, connectionism versus symbolic AI being the best-known example. There are incompatibilities on a smaller scale too, for example the split between classical and non-representational or behaviour-based AI. It may be that, like architecture, AI can be no more than a collection of tools and techniques. Or it may be that there can be found some unifying concepts which underly the whole of the subject.

5 AI as engineering versus AI as psychology

The first is concerned with making intelligent machines, not with building accurate models of mind. However, we can learn a lot from biological systems. A lot of different tools are available, some of which I'll talk about in later lectures. To the outsider, its best-known accomplishments are expert systems and neural nets.

The second raises many questions which the engineer can safely ignore, particularly about exactly what one is modelling.

6 The design stance

A good way to look at AI as psychology. The idea that to understand how a natural intelligence works, one tries building a working replica. Once we have several going, we can ask which of them (if any) is most likely to be embodied in the brain, and how. This helps suggest psychological and neurophysiological experiments.

Probably most successful so far in low-level vision, where we have fairly tight constraints on the task, the hardware available, and the structure of the information available (geometry of 3-d space, physics of light, rigidity of most objects, etc).

ILP is backed by principled statistical methods. It also has the great advantage — as a tool — that the results of its learning are easy to comprehend. Incidentally, AI will become increasingly important in pharmacology. Worth bearing in mind if you intend to make a career in AI.

7 Symbolic versus connectionist AI

I shall deal almost entirely with symbolic AI in my lectures.

The name "symbolic AI" is usually opposed to "connectionist" AI, also known as PDP or neural nets. However, terminology differs between authors.

For example, according to Chandrasekaran, Fodor and Pylyshyn talk about "classical" models, Dennett uses the term "High Church Computationalism", and so on. But the term "classical AI" is not well-defined either. Some authors use it to refer to "symbolic AI" combined with assumptions about the need for explicit representation of world-models, goals, and such entitities. See page 45 in *Intelligence as Information Processing*, in *Foundations of Artificial Intelligence* edited by Partridge and Wilks (PSY KH:P 025). This article analyses the difference between various AI paradigms, and between differences in terminology. I recommend reading it, preferably after you've seen examples of both symbolic AI and connectionism.

So if you use such terms, define what you mean! Strictly, "symbolic AI" is a misnomer, because connectionist systems also symbolise things. It would be better to say "AI as computation over discrete symbol systems". I shall explain why in the following section. However, for brevity, I'll stick to the term "symbolic AI".

7.1 Symbolic AI — symbolic manipulation systems

You may want to skim these sections on first reading.

Most of the AI I describe this term is "symbolic AI" in the above sense, including the memory models (lists, semantic nets, Schank); rule-based systems (expert systems, production systems and Soar); and miscellaneous examples such as Evans' Analogy Program and GPS. Beer's cockroach and Cliff's evolving robots are not.

In symbolic AI, we assume there to be a fixed set or *alphabet* of elementary symbols which is known in advance. These symbols may be combined in various ways, but not all combinations are allowed. So we need some rules of *syntax* to specify which combinations are valid. This gives us our basic notation or "language of thought". (We don't need to assume that an intelligence only uses one notation: PopBeast and many other programs use more than one.)

The alphabet is finite, and arbitrary crosses between elementary symbols are not allowed. For example, there is nothing between an O and a P. Hence "discrete" in the previous section. Furthermore, the symbols are not analogue — there's nothing in the shape of a symbol which relates it to the thing it symbolises.

As well as syntax, we have also rules of *semantics* which specify how the meaning of these combinations depends on the meaning of the component symbols. These syntax and semantics rules need not themselves be explicitly present. If they were, we'd have to encode them too, presumably in some other notation, and we'd hit an infinite regress. Instead, the rules of syntax and semantics emerge from properties of the "hardware". Actually, this isn't quite true. As a later section shows, you can have hierarchies of symbol system, and the syntax and semantics of a higher level can be encoded in a lower-level system. But ultimately, this descent stops at the hardware.

Finally, there are rules for manipulating these symbol combinations, which derive new combinations from old. The symbol manipulation system is no use unless it "runs" or "computes": these rules might be hard-wired into some hardware, or might be software — a program.

Example of symbol manipulation system: decimal arithmetic. Alphabet contains the digits, brackets, plus, minus, etc. Syntax specifies that (for example) `12(*` is invalid, but `12+3*(45)` is valid. Semantics specifies the meaning of addition, the fact that the `1` in `123` is ten times more than that in `12`, and so on. The rules of manipulation tell us that we can replace `123+4` by `127`, and so on.

7.2 Symbol manipulation systems in AI as engineering

For AI as engineering, this is all straightforward. Many AI programs are written in the languages Lisp or Prolog, and the elementary symbols will usually correspond to numeric and string constants. In Prolog, they'd include what are called "atoms". You should look at a few examples: see for instance the description of Evans' Analogy program in pages 24–32 of *Artificial Intelligence* by P Winston (2nd edition Addison-Wesley 1992: PSY KH:W 072).

These references do not explicitly state the symbols, syntax rules, and so on, but you can easily get some idea of what they should be. In all three references, at least some of the notation rather resembles logic. This is no accident.

7.3 Symbol manipulation systems in AI as psychology

Here, this is much less clear. Firstly, it's only a hypothesis that all or part of the mind is (or can be modelled as) a symbol manipulation system. Assuming it can, does, how are the symbols and the rules of manipulation encoded?

Most practitioners assume that, as long as we can find out what the symbols and rules are, we need not worry about the encoding. The mind can be studied at a purely symbolic level, without having to delve into the hardware. In particular, we can map the symbols and rules into a computer, and hence create a program which solves the same problems in the same way as the original.

So a symbolic-AI theory of some mental process must include

- Firstly, a description of the elementary symbols, their allowed combinations (syntax), and the way the meaning of these combinations is determined (semantics).
- Secondly, the operations which exist for manipulating these combinations.
- Thirdly, many models will place restrictions on the complexity or size of the symbol combinations. Strictly speaking, this is part of the syntax, but it's often stated separately. The most obvious example is the capacity of STM in memory models.
- Fourthly, it may also try to give an idea of how and where the symbols and rules are encoded, and of timing: how long do the basic operations take, and do they occur in serial or parallel? Most theories say very little about neural encoding.

Altogether, these constitute the *functional architecture* of the part of the mind being modelled. The name comes from computing, and implies that one is ignoring lower levels of detail. For those who know a bit about computers, one might talk about the functional architecture of a 486 PC: it has a linear array of memory cells, each capable of holding 32 binary digits; these digits can be treated as numbers, and added, multiplied, etc. There is a limit on the size of memory, usually between one million and ten million cells. There are some elementary operations for manipulating these numbers; these are also represented as 32-digit sequences, and are also stored in memory. Like the numbers, they can be operated on and changed. Only one number can be operated on at a time, and this must be done by copying it from memory into an accumulator. Addition takes this number of nanoseconds, multiplication takes that number of nanoseconds, and so on.

For a psychological example of functional architecture, see page 164 of *Computer Models of Mind* by Boden (CUP 1988; PSY KD:B 063), Also, read the Young article on *Production Systems for Modelling Human Cognition* in *Expert Systems in the Microelectronic Age*, edited by Michie (E.U.P. 1980; PSY KH:M 58). Quite apart from the idea of functional architecture, PSs are important as developmental models, and there'll probably be a question on them.

For more on the symbolic-AI approach, see the summary of "High Church Computationalism" in section 1.2.2 of *Intelligence as Adaptive Behaviour*. Beer quotes Dennett's summary of those shared assumptions behind the symbolic AI approach. Note: I am teaching you about this approach, but not necessarily agreeing with it all. Note also that there is much debate about what constitutes a sumbol, etc.

7.4 Hierarchies of functional architecture

In computing, we could implement C in machine-code, write a Lisp compiler in C, and then write an analogy-solving program in Lisp. Each of these systems defines its own set of symbols, syntax rules, semantic rules, and operations. Think of any program, e.g. Wordperfect: it gives you a basic set of capabilities which are quite unlike those of the PC's machine code. So we can also describe the *functional architecture* of each higher level: of C, Lisp, etc. These higher levels are often called *virtual machines*: "imaginary" machines implemented as software, not hardware. There's a very nice discussion of this in Chapter X of Hofstadter's *Gödel, Escher, Bach* (Penguin 1981: PSY KH:H 67).

Engineers like neat hierarchies because it makes thinking about the problem easier. It's an open question as to whether the brain works like this, and indeed whether any complete intelligence can. Perhaps, for true intelligence, the levels must all be horribly mixed: see discussion of evolved robots in my second AI and PP lecture.

7.5 Vague models

Most symbolic AI models don't specify their functional architecture in much detail, and do not state exactly which features are accidental to the theory. Young, on page 43 of *Production Systems for Modelling Human Cognition*, discusses this further.

One example of such a theory is in *Dynamic memory* by Roger Schank (CUP 1982; PSY BG:S 299). Definitely read this. You can understand the first two chapters without any technical knowledge at all, and these will give a good idea of symbolic AI applied to reminding and learning. I shall cover some of this work in my second AI lecture.

How should AI papers be evaluated? What's the worth of these models? One researcher argues that the objective of AI is to suggest approaches: to show that, in principle, a given kind of architecture can be made to solve a given class of problems, and then to leave psychologists to refine this. See pages 213–218 of *Vision, Instruction, and Action* by Chapman (PSY KH:C 036). This comment could apply to all AI as psychology, not merely symbolic AI.

7.6 The importance of representation

If our notation and rules of manipulation are limited, some things will be easier to express than others, and some manipulations will be easier to perform than others. Many familiar examples: arithmetic in Roman vs Arabic numerals; finding words or meanings in a dictionary vs a thesaurus; different types of filing system, whether paper or computerised; random access to casettes vs CDs; DNA is easier to copy, but protein is better for controlling synthesis; the effect of cutting up a photo vs a hologram; keeping track of position in a queue by holding a numbered ticket vs by standing in order; calculating the number of days between two dates written in d/m/y vs day-number notation; calculating the month of a date written in d/m/y vs day-number notation.

In general, a good representation allows you to express the information needed for the task being performed, in such a way that it can easily be processed by the operations you have available. See page 24 of Winston.

What has this to do with the way we should explain the workings of natural information-processing systems? See *Vision* by Marr, pp 15–38. (W H Freeman 1982: PSY CH:M 34; RSL Physiol M 12 b).

8 Applications of AI

Now some examples. These cover both engineering and psychology, and illustrate various principles.

8.1 Helping motorists from police reports — Poetic

A typical example of a symbolic AI application. "Portable Extendable Traffic Information Collator". A natural-language system which reads police incident logs and selects out information about those traffic problems which might concern other motorists. From these, builds up a coherent description of each incident, and then works out how motorists in the area might avoid the problem. Converts the advice into English, and then decides where best to broadcast each message. See *POETIC: A System for gathering and distributing traffic information* by R Gaizauskas, R Evans, L J Cahill, J Richardson, J Walker, photocopy G87 in the AI box: these include examples of the police logs that serve as input, and the advice generated as output, and are easy to read even if you are new to AI.

Some points about Poetic:

- One thing it must do is to build up a *world model*: a coherent description of the world, derived from its sensory input. It's been a common assumption in AI (now being challenged, see footnotes in my first tutorial handout and the PopBeast stuff with it) that this is a necessary part of any intelligent organism, and that animals and humans will embody such a model. Like many other symbolic AI programs, Poetic's world-model is in logic, in this case via the Prolog programming language.

- This may involve cross-referencing different log entries to disambiguate references to objects. For example "8.05: lorry crash on A413" and "9.17: road is still blocked". Are these the same road? Illustrates a general problem: perceptions are usually incomplete. If an agent's world model is going to represent individual objects, the agent will have to match up different partial descriptions to determine whether they refer to the same object. These descriptions may come from different sensors at the same time, or the same sensor at different times.

- The natural-language understander must cope with ungrammatical input and jargon. Police logs are not standard English! Also, not all the information in the logs is about traffic problems, so Poetic must analyse the entries before it knows whether to ignore their content.

8.2 Game playing

Chess players: familiar to all as the Chess Challenger, etc. A lot of progress from big programs: Belle in 1983 got US Master status; Deep Thought in 1989 got a US Chess Federation rating of 2500, aspiring Grand Master status. See article on *Computer Chess* in the *Encyclopaedia of AI*.

Backgammon: Hans Berliner's program defeated the world champion Villa in 1979. There have been a lot of improvements since. See *Computer Backgammon* by Berliner, in the June 1980 Scientific American. [4]

Go: more intelligent than chess programs. The latter rely on brute-force search and fast computers. Most chess programs search the tree of possible moves to five or six levels deep. Eventually, the branching factor will overwhelm them: an example of *combinatorial explosion*. But Go has many more possible alternative moves from any position — perhaps 150–200 options as against 25 for typical mid-game chess. So Go programs can only search one level deep. This means that they must consider alternative moves, and evaluate each in terms of global strategy. Hence — say many Go players — a successful Go program will probably "see" the board in terms of eyes and other patterns, and react to these as an expert would.

As Berliner points out, the same applies to backgammon, and his article is a nice introduction to the subject of intelligent searching.

So how do humans play games? Concerns AI-as-engineering because game-playing machines sell (and the methods might apply to other kinds of problem); concerns AI-as-psychology just for the intrinsic interest of knowing the mechanism.

De Groot: one of the earliest to ask this question. Chess masters and weaker players differ in performance — what's the difference in mechanism? Difference in search methods perhaps: depth of search, or number of alternatives considered for each move? Experimental evidence suggests not: de Groot found these the same for masters and weaker players.

But although masters and weaker players consider the same number of moves, masters consider *better* moves! Also, masters are better at remembering (correct) chess positions. Instead of perceiving them as collections of individual pieces, they appear to work in larger chunks, such as pawn chains, castled-King positions. These units are probably strategically significant, and may serve to trigger suitable moves for further consideration. [5] See article on *Human Chess Skill* by Neil Charness in *Chess Skills in Man and Machine* edited by Frey (2nd ed 1984; PSY KH:F 089). [6]

Several important themes of AI arise from this:

- The interplay between the engineering and cognitive modelling sides of AI.
- *Combinatorial explosion*. Search is expensive. It takes time to do, and space to store the alternatives. In most problems, it will be forever impossible to build computers that are fast enough to search through all possible alternatives.
- *Evaluation functions*. In search, whether we consider all alternatives, or only some, we have to know when one is good enough. This involves trying to evaluate it against other alternatives. The evaluation must be done quickly, and may only be approximate.
- *AI as productive laziness*. A slogan from *Computers and Thought*, by M Sharples, D Hogg, C Hutchison, S Torrance, and D Young (MIT 1989; PSY KH:S531; RSL M92.E00068; Hooke).

  How can an agent apply its knowledge to cut down the number of possibilities it must search through? What's the best way to store that knowledge? Can clever indexing methods reduce the time taken to find items?

[4] There's also an article by Berliner in the *AI Journal* volume 1 for 1980, RSL open shelves.

[5] For example, a position might be perceived as "queen on open QR2–KN8 diagonal; knight can reach KB7 in one move; opponent's king on KR1 is hemmed in at KN2 and KR2". Features 1, and 2 or 3, might trigger several possible countermoves, through associations built up by experience.

[6] This article describes the most famous cognitive model of chess skill. The main references to it are *Perception in Chess* by Chase and Simon, in *Cognitive Psychology* volume 4, pp 55–81 (1973). *A simulation of memory for chess positions* by Simon and Gilmartin, *Cognitive Psychology* volume 5, pp 29–46 (1973).

- *The paradox of the expert*. Surely as the amount of knowledge increases, the time taken to find an item should take longer. Not in people! Perhaps this is because we can reconfigure our representations, e.g. by changing the size and type of chunk (going from letters to words, or chess pieces to positional features). As yet, we have very little idea of how to make a program do this, or of how humans avoid the paradox. [7]

8.3 Automated grammar checking

Most word-processors have spelling checkers. But could we make something that detects and corrects grammatical errors? This could help spelling checkers too — how else could such a program know whether to substitute "their" or "there" in the sentence "ther are four chairs here."?

There are several grammar-checkers already available, though most have a very sketchy coverage of the language. I worked on one such program, the "Editor's Assistant". One problem that arose: "morphological generation". If the program encounters the third person present plural of a verb, and decides it should be corrected to third person past plural, how to generate the correct word-form? The lexicon has to contain information about which verbs are irregular, and how to inflect them. Luckily, English has few inflections, and despite oddities like "mother(s)-in-law", "child(ren)", most are comparatively regular. Some other languages are much less regular. All this raises the question of how to index the lexicon in such a way that the right entry can be rapidly located and modified, given its root form and a list of grammatical features. Such questions of indexing and storage — how to represent information so it can be used efficiently — are a central part of AI, as mentioned above.

More speculatively, how about an "intelligent" editor that can be told "find all references to Queen Harriet in this story I'm writing, and replace them by King Harry, changing the relevant pronouns where necessary. Warn me if I've mentioned details of dress etc that must be changed because they depend on sex." The Editor's Assistant could actually cope with a similar but simpler task, which involved finding explicit references to measurements in miles, and replacing them by the equivalent number of kilometers.

Such an editor would require a vast amount of knowledge about the real world. Again, how do we encode it so that it can be efficiently located? And is there *any* formalism which is general enough to say everything that needs to be said? [8]

A need for such knowledge may crop up in unexpected places. The following example comes from Henry Thompson, a language researcher at Edinburgh. Consider translating the sentence "The Council committee members banned the womens' parade because they feared violence" into French. Does "they" become "ils" or "elles"? Now replace "feared" by "advocated". It seems that our understanding of this depends on the real-world knowledge that, sexist though it be, Council committees are usually largely male. There are various theories of language understanding which call for the use of such knowledge at various stages of analysis. When is it brought in? How does the analyser know what it needs to look for?

[7] Two pieces of research that suggest how this may one day be done. *Analogy-making as perception* by Mitchell (MIT 1993; PSY KH:M 069). And *Declarative learning* by Furse and Nicolson in *Proc 14th Cog Sci Soc* (PSY BH:C 065) and *Mathematical expertise and the contextual memory system* by Furse (AI photocopy F57). To me, the second seems particularly promising, as a method of organising what you learn without making preconceptions about which features to concentrate on.

[8] The sole attempt at this: *Building large knowledge-based systems* by Lenat and Guha (Addison-Wesley 1990; PSY KH:L 054).

8.4 Robotics, and vision

Many applications here — imagine a robot vacuum cleaner, for instance. But robotics really demonstrates the failures of AI. Mike Brady, the head of the Oxford Robotics Group, declared in a public lecture two years ago that he would consider his life work done if he could design a robot that was as competent as a sheep! Although AI has been quite successful at "rational thought" tasks like language analysis and game-playing, it is much less so for tasks that involve perception and action. Although our TV cameras can provide all the information a robot might need, we are nowhere near being able to use it in a general-purpose vision system. Much of robotics involves designing sensors to compensate for the limitations of robotic thought: for example, by projecting grids onto the scene to help locate objects.

Because classical symbolic AI has been so unsuccessful here, many people are trying alternative approaches. [9]

8.5 The computational cockroach

An example of AI as (animal) psychology. Which, incidentally, demonstrates that AI is not concerned solely with modelling or implementing theorem-proving, game-playing, and other instances of "rational thought". Nor is it concerned solely with human abilities.

This example is from *Intelligence as Adaptive Behaviour* by Beer (RSL M91.E00889; PSY KH:B 039). Beer builds a computer-simulated artificial cockroach in order to understand the neural mechanisms underlying some of its basic behviours: walking, wandering, recoil from objects, feeding. But his main interest is the mechanisms underlying the choice between behaviours (page 144). In some cases, one controller may supress or enable another directly by inhibitory or excitatory connections. However (page 152), matters aren't always so simple. One important point of Beer's work is that we can't understand the neural basis of such behaviours without also understanding the dynamics of the body (page 169).

In the work I talked about, Beer starts with Pearson's work on the neural basis of cockroach walking (page 73). Pearson proposed a model for how each leg is controlled, and a suggestion about how control of different legs is coupled. Beer built a computer implementation of that model and found that it correctly predicted fast (tripod) gaits, but not slow ones (page 78). So Beer incorporated an idea — phase-locking — from a more general study on insects. This did predict the slow cockroach gaits, though not those of some other insects (page 81). In summary, Beer implemented an existing model, discovered that it was insufficient, and demonstrated some plausible additions that made it more predictive. Implementing a theory on the computer forces one to consider details that might otherwise be neglected.

8.6 Sloman's model of memory for sequences

An example model of a simple cognitive task, and of the style of explanation AI offers. Why is it easier to recite the alphabet forwards than backwards? **Note** Neither Sloman nor I are proposing this as a serious cognitive model, merely as an example of a style of explanation.

A mechanical analogy: we decide, for no good reason, to store the letters of the alphabet in a filing cabinet. To store `A`, we write it on a card, find an empty drawer (say 8), and put the card inside. Make a note of this as the starting position. Later, we store a `B`. We pick another drawer at random. Say it's 15. We put the card in 15, and then go back to drawer 8, writing 15 on `A`'s card. Later still, store `C`. We find

[9] See *An emerging paradigm in robot architecture* by Malcolm, Smithers and Hallam, AI photocopy 149.

another drawer — 1 say — put the card for `C` there, and then go back to write 1 on `B`'s card. And so on, to `Z`, which goes into drawer 435.

Now, consider somebody who's asked to start at `A`, and recite the alphabet forward. This is easy. Start at 8, pick the card; find the drawer it mentions, and repeat. Going backwards is harder…Start at 435, and then what?

I could, using what computer-scientists call "singlely linked lists", build a robot which stored sequences in an analogous fashion. Such a robot would find it much quicker to recite forwards than backwards. Alternatively, I could build one where each cell (card) bears two numbers, one pointing at the previous cell, and one at the following cell. Now there'd be no difference between the two directions. So the way we represent information crucially affects its processing.

This example comes from *The Computer Revolution in Philosophy* by Aaron Sloman (1978). In the Bodleian (one copy as 26684 e 1106 and one as 266 e 603 unless reclassified). Chapter 8, especially sections 8.1, 8.3 - 8.10, and 8.16, is an account of the kind of representation that might build up in a child learning to count. This is Sloman's view of how cognitive psychology ought to answer questions like "Why is it easier to count forwards than backwards?", and it is typical of the AI approach to cognitive modelling.

9 Semantic nets — memory for hierarchies

Now something which actually was proposed as a model of memory. Taken from *Cognition* by Posner, pp 86–89 (PSY BH:P 084), and *Computational Semantics* edited by Charniak and Wilks, pp 198–200 (PSY KH:C 038). Both these show a diagram of the hierarchy. Note that further experiments showed the model to be invalid.

From the point of view of the design stance, we see that a rational designer would try optimising storage (storage space is always limited). One way to do so is to store the properties as few times as possible. So instead of tagging each bird with the property "has wings", we just add the property once, to the topmost appropriate node.