Chapter 3

Artificial Knowledge

3.1 Introduction

This chapter moves on from the metaphysical consideration of what kind of a thing a mind is, or might be, to the pragmatic consideration of building machines (especially computers) that exhibit some or all of the *behaviours* associated with mentality—which is to say, a consideration of Artificial Intelligence (AI) in what Searle (1980) calls the "weak" sense. Alternatively this may be viewed as an investigation of the hypothesis I have previously (Chapter 2, section 2.2) called *Turing Test Computationalism* (H_t)—the claim that a suitably programmed universal computer could pass the Turing Test.

I start with a brief review of the Turing Test itself, and, in particular, some novel criticisms of it proposed by French (1990). I shall consider these criticisms, but argue that the Test still stands as a valuable focus for work in AI; nonetheless, I shall go on to conclude that performance at this level is still so far beyond our present theoretical understanding, that Turing Testing, as such, may of little immediate or practical interest.

I next consider the general issue of *cognitive architecture*—what, if anything, can we say about the overall structure which a (computational) system must have if it is to exhibit behaviours indicative of intelligence. The essential point I make is the negative one that *universality* (in the technical sense characterised by, for example, Universal Turing Machines), *per se,* does not mean that a computational intelligence will admit of explanation in terms of a unitary “symbol level”, or “language of thought”.

I then consider the notions of “meaning” and “knowledge” in more detail, in an effort to show that a *computational semantics* is indeed possible (despite some claims to the contrary), and I sketch out what it might look like. In particular, I claim that computers can realise *anticipatory systems* (Rosen 1985a), and that, in this case, they exhibit *intentionality* (Dennett 1978b), and instantiate *subjective knowledge* in the sense that Popper admits for biological organisms generally (e.g. Popper 1961). These claims are made *independently* of any commitment to the idea that computers are able to realise “genuine” mentality—in the sense of “conscious subjective experience”.

With this particular philosophical perspective, I then briefly consider methodological approaches to AI, in particular the notion of “Knowledge Engineering”. I note that this approach has run into serious difficulties, typically identified with the *common-sense knowledge* problem. It has proven extremely difficult to *explicitly* formulate common-sense knowledge (and thus incorporate it into computer systems). There is little general agreement as to the nature of this problem; but it seems that developing an explicit, brute force, *stipulation* or *enumeration* of common-sense knowledge is currently an intractable problem, and may yet prove to be completely impossible.

The alternative to the Knowledge Engineering approach is, of course, to develop some kind of “adaptive” or “learning” system; which is to say, we turn from the problem of knowledge in itself, to the rather different problem of its *growth*.

I shall argue, from several different points of view, but based particularly on the *evolutionary epistemology* pioneered by Popper and D.T. Campbell, that a kind of abstract generalisation of *Darwinian* processes, referred to as *Unjustified Variation and Selective Retention* (UVSR), is an essential component in the growth of knowledge. I conclude from this that the realisation of *Artificial Darwinism* may a necessary, though certainly not sufficient, condition for the realisation of Artificial Intelligence.

3.2 The Turing Test

3.2.1 Definition

In his influential paper, *Computing Machinery and Intelligence* (Turing 1950), Alan Turing set out to consider the question "Can machines think?" (p. 433); ultimately, however, he concluded that, in this form, the question was "too meaningless to deserve discussion" (p. 442). Instead, Turing proposed an *operational* definition for "thinking", and restricted "machine" to designate a suitably programmed *digital computer*. He then considered the new question of whether such a machine could satisfy such a definition.

This operational definition of thinking was phrased in terms of what Turing called the "Imitation Game", and is now generally referred to as the *Turing Test*.

Briefly, the Turing Test involves a human *interrogator*, and two *subjects*. One subject is the *machine* to be tested, the other is a human *control*. The interrogator is restricted to interacting with the two subjects purely linguistically (for example, via teletype), and has no other way of distinguishing between them. One *turn* then consists of a fixed time—Turing suggests 5 minutes—in which the interrogator is allowed to question the subjects, after which time he must nominate which subject he judges to be the human and which the machine. The machine is considered to have *passed* the Test if the interrogator's *probability* of making a successful classification is found to be below some specified threshold—Turing suggests 70%. Turing omitted various details here: one presumes that the success probability would be measured by playing out as many turns as are necessary to get a statistically significant result, while varying the interrogators and control subjects to achieve independence between turns. Turing does explicitly refer to the use of an "average" interrogator (p. 442).

3.2.2 Sufficiency?

As discussed in Chapter 2, there is room for argument as to the *sufficiency* of the Turing Test. That is, whether an entity's ability to pass this Test is a sufficient condition for saying that it exhibits *mentality*. If the Test were not sufficient in this sense then that would certainly limit its interest. However I have already stated, in Chapter 2, my opinion that the proposed arguments to such an effect are far from compelling; and that I shall therefore proceed on the basis that, *pro tem,* Turing Testing *is* a sufficient operational criterion for mentality.

3.2.3 Necessity?

French (1990) takes the position that the Turing Test is valid or sufficient for the attribution of intelligence, but argues that it is in fact much *more* stringent than Turing anticipated or intended. Specifically, he suggests that "... the Turing Test could be passed only by things that have experienced the world as we have experienced it" (p. 53). While he believes that *in principle* a machine could indeed be built which would satisfy this constraint, he assumes that, *in practice,* "no computer is now, or will in the foreseeable future be, in a position to do so" (p. 56). It follows, of course, that there is no practical prospect of a computer passing the Turing Test. French concludes that some alternative tests, or at least criteria, are therefore needed for practical use in AI research.

I should emphasise that I agree with French on certain points which he raises. For example, he suggests that the Turing Test is deficient in that it admits of no "degrees" of intelligence, and is not applicable at all to non-linguistic behaviour that might, in fact, be related to intelligence (such as exhibited by animals). I agree with this as far as it goes: given that Turing Test performance is currently an intractable problem, it is sensible to formulate lesser or entirely distinct criteria which might, once achieved, represent progress toward that ultimate goal. In fact, this is what goes on in practical AI research all the time.

Where I disagree with French is when he goes on to suggest that the Turing Test should be dispensed with altogether, even as an *ultimate goal* against which intermediate goals can and should be critically reviewed. Even here, I shall give some ground, though not, I think, as much as French seeks.

French's argument is that the Test, as formulated by Turing, admits the use of so-called *subcognitive probing* by the interrogator, and that this makes the procedure an unnecessarily *harsh* or severe test of general intelligence. That is, French supposes that there could be systems (presumably including certain suitably programmed computers?) which would be unable to pass the Turing Test, but which should, nonetheless, be labelled intelligent—indeed, "as" intelligent as humans, if not more so.

The idea of subcognitive probing is to ask questions which, in some sense, probe the underlying, subconscious, "structure" (the *associative concept network*[1]) of the putatively intelligent subject. French argues that this is possible under the Turing Test conditions, and that it would allow specifically or uniquely human aspects of intelligence to be detected—aspects which would be very difficult, if not entirely impractical, to duplicate in a computer, and which are, in any case, *inessential to general intelligence.*

In fact, French concludes that the practical development of some entity such that it could pass the Turing Test, given the use of subcognitive probing, would require that the entity be capable of experiencing the world "in a manner indistinguishable from a human being—a machine that can fall off bicycles, be scratched by thorns on roses, smell sewage, and taste strawberries..." (French 1990, p. 56): that is, the system would have to be a more or less humanoid robot or *android.* It is this scenario which French regards as being impractical (though not, in principle, impossible) for the foreseeable future. More to the point, he considers that this renders the Turing Test unsuitable for practical use.

French further claims that the Turing Test cannot be modified, in any reasonable way, so as to eliminate the possibility of subcognitive probing, and should therefore be simply discarded. He does not propose a specific, operational, alternative, but suggests that we should consider intelligence in the "more elusive terms of the ability to categorise, to generalize, to make analogies, to learn, and so on" (p. 65).

I agree with French that the use of subcognitive probing, as he describes it, would subvert the Turing Test; that, indeed, such probing is one of the general kinds of thing Turing was trying to preempt in his design of the Test; and that only some test which does not exploit such probing would be satisfactory. However, I disagree with French that such probing cannot be eliminated from the Turing Test, with little or no modification. I shall argue this on several grounds.

[1] French presumes that some such network necessarily underlies intelligence; I do not disagree as such, but it might have been better if he had made his assumption explicit, and phrased it as an *hypothesis,* rather than taking it as some kind of established fact.

First, and most obviously, French is able to introduce subcognitive probing in the first place only by effectively changing (or, at least, augmenting) the rules of the original Test. Specifically he requires that the interrogator be allowed to *poll* humans for the answers to some questions prior to posing them during the Test itself. This is in order to allow statistical analysis of the "subcognitive" characteristics of responses to these questions, as exhibited by people, so that these could then be compared with the behaviours of the subjects in the Test proper. French states that he feels "certain" that Turing would have accepted this. I happen to disagree with this opinion, but it is irrelevant in any case. The point is that if we *disallow* such polling (whether Turing would have approved or not) the Test is effectively immunised against the use of subcognitive probing, by French's own admission.

But quite aside from this, I think French's analysis is contrived and mistaken. While Turing did not specify precisely what he meant by an "average" interrogator, it seems absurd to suppose that he would have allowed interrogators who are familiar with, and competent to consciously apply, the notion of subcognitive probing. Again, of course, the question of what Turing's own opinion might have been is strictly irrelevant anyway: the important point is that, in response to French's criticism, we are quite free to add an explicit stipulation to the Test, to the effect that persons having a competence in the technique of subcognitive probing will not be allowed as interrogators—*if* that is deemed necessary in order to eliminate subcognitive probing. In fact, I suggest that, for virtually any practical purposes, it would be adequate simply to *stipulate* to interrogators, at the start of any Test, that they must not attempt to *use* subcognitive probing in their evaluation of the subjects.

French might still argue for the possibility of *unconscious* subcognitive probing having some statistically significant effect on the Test outcome. This would obviously be, at best, a much weaker argument, and I don't believe it could be sustained in any case. Remember that the Test, as Turing specified it, is relatively coarse (presumably deliberately?): the interrogators' success rate only has to fall below about 70% for the computer to pass. I doubt very much that a credible argument could be made to the effect that subcognitive factors, *alone,* are likely to consistently, and unconsciously, bias the success rate by 30 percentage points or more.

But even if, despite its intuitive implausibility, we suppose that French could marshal enough evidence to show an effect of this magnitude due solely to unconscious subcognitive factors, I claim that the effect could *still* be nullified with relative ease. This can be done by interposing what I shall call a *subcognitive scrambler* between the interrogator and the subjects. This would simply be another person, who relays all messages between the interrogator and the subjects. The interrogator is now restricted to have direct access only to the scrambler, and not to the subjects. The scrambler takes up the previous position of the interrogator, having linguistic access to the subjects, via teletype or otherwise, but otherwise having no knowledge of the identities of the subjects. The sole instruction to the scrambler is to *paraphrase* the messages passed from interrogator to subjects, and back, in such a way as to maintain their essential semantic content, but to otherwise modify them as much as he wishes. A particularly effective way to achieve this might be to use interrogators whose native language is different from that of the subjects, and thus have a *translator* act as the subcognitive scrambler.[2]

I freely admit that such a scrambler would not be effective against *all* kinds of deliberate or conscious attempts at subcognitive probing.[3] However, I think it would greatly attenuate any possible *subconscious* subcognitive effects, which was the remaining point at issue.

In conclusion then, I consider that the deficiency in the Turing Test, alleged by French (i.e. its supposedly *excessive* stringency), is either non-existent or easily corrected, and the Test can therefore survive his attack more or less unscathed.

[2] In allowing, or even recommending, the use of such translation, I implicitly transgress, to at least some extent, against another assumption which French allowed himself: that the human subject and the interrogator "are all from the same culture and that the computer will be attempting to pass as an individual from that culture". Again, I see this as *ad hoc* and contrived on French's part, and not sustainable.

[3] I have in mind specifically what French calls the *Category Rating Game* technique.

3.2.4 An Informal Test

And yet: while I disagree with French's literal arguments, I cannot help but believe that there is some core of truth about his ideas.

Let me suggest then that detailed, legalistic, discussion of the Turing Test is pedantic, and essentially futile—notwithstanding the fact that I have just indulged in such a discussion above. I indulged in it because that is the ground on which French had chosen to mount his assault, so I wished to respond in kind; demonstrating that, judged even in his own terms, his assault founders. However, in many ways it was a pity that Turing gave a relatively precise description of his proposed Test—for it is this spurious precision that prompts excessive concentration on the details, such as exhibited by French.

I suggest that the Turing Test should best be considered as a blunt (though moderately effective) instrument, whose details are entirely unimportant. Its point lies not in any detailed experimental set up, but in the *principle* that any machine which can credibly, or meaningfully, participate in human conversation should, *regardless of what other attributes it may have* (especially its physical constitution), be regarded as a *bona fide* member of the community of sentient beings.

I suggest especially that *indistinguishability* between machine and human conversation, which is at the core of much discussion of the Test, including that of French, is actually a red herring. I think that this is implicit in the rather coarse tolerance of 70% originally suggested by Turing for his Test.

The real issue is *credibility*: whether some putative machine intelligence can sustain a conversation in such a way that we would be satisfied that it really *means* what it says; this remains the case, even if what it is saying is obviously and thoroughly non-human (and thus perfectly "distinguishable" from human conversation). For example, the conversation that would be involved in actually inviting a machine to act as a subject in a formal Turing Test would certainly involve elements that would not arise in any normal conversation between human beings; but I suspect that, on the basis of just such a conversation, one could sensibly judge whether the machine *meant* what it was saying or not.

So, if French's point is that the Turing Test, as stated, focuses on indistinguishability from strictly human intelligence, and that this is unnecessary and even misguided, then I am inclined to agree with him. French however, sees this as an *intrinsic* defect of the Test. I think he is mistaken in this, as I have already argued; but even if he were right, I think this conclusion would be contingent on a very literal reading of the Test (which, I admit, overemphasises the issue of comparing machine with human intelligence), and a consequent failure to appreciate the central, informal, idea being promoted by Turing.

What I take to be the proper view of the Turing Test has been previously elaborated by Roger Penrose:

> From my own point of view I should be prepared to weaken the requirements of the Turing test very considerably. It seems to me that asking the computer to imitate a human being so closely so (*sic*) as to be indistinguishable from one in the relevant ways is really asking more of the computer than necessary. All I would myself ask for would be that our perceptive interrogator should really feel convinced, from the nature of the computer's replies, that there is a *conscious presence* underlying these replies—albeit a possibly alien one. This is something manifestly absent from all computer systems that have been constructed to date.
>
> Penrose (1990, p. 11, original emphasis)

To be clear then, let me now propose what I shall dub the *Penrose Test* for intelligence:[4]

> Any entity is intelligent which understands and means what it says; and any entity understands and means what is says if it can so convince a competent human judge, purely on the basis of her conversation with it.

By a "competent" judge I mean someone who, *inter alia,* has a reasonable understanding of the state of the art in AI, and is capable thereby of probing past "canned" responses of the so-called `ELIZA` type (Weizenbaum 1984), etc. Indeed the judge should probably have some specific familiarity with whatever design principles may have been used in building the putative intelligence (in this limited respect, the test I propose here is arguably *more* stringent than Turing's).

[4] Perhaps this might equally be called the *Asimov Test* for intelligence; compare it with this formulation: "There is no right to deny freedom to any object with a mind advanced enough to grasp the concept and desire the state" (Asimov 1976, p. 174). The "grasping" and "desiring" are apparently to be established by similar criteria to those which I have suggested: linguistic cross-examination of the subject. In Asimov's case a "competent" judge is, effectively, any court of law having relevant jurisdiction.

I have omitted any comment about the allowed or required *domain of discourse* in the Penrose Test. This is deliberate. I consider that the demand that the entity convince a competent judge, purely through conversation, that it really does understand and mean what it says is already enough to guarantee a satisfactorily wide ranging domain of discourse, without any additional stipulation.

My claim is that the Penrose Test captures the essence of Turing's original Test; and that, in particular, any honest researcher can judge perfectly well whether his system should be labelled intelligent, in this sense, without ever having recourse to the elaborate paraphernalia actually prescribed by Turing, and without any significant danger of being confounded by irrelevant factors, subcognitive or otherwise.

Furthermore, I suggest that this is in fact the way the "Turing Test" is employed by practical researchers. I think it is generally accepted that no AI system yet developed has come remotely close to meeting Turing's criterion, and this is known without any attempts at setting up the kind of formal test conditions actually described by Turing. The latter would only come into play if or when we have a system which we *already* know, from the Penrose Test, to have the depth of understanding required to participate in a meaningful conversation; but even then the formal Turing Test would, at best, serve only to demonstrate the objectivity of this claim. And of course, we should remember that the rôle of the machine in the Turing Test is distinctly demeaning, if not positively insulting: it seems to me that a *prima facie* mind might well refuse to participate in such a charade!

For a more detailed discussion of the issues arising here, see Hofstadter's (1985, Chapter 22, Post Scriptum) account of actually attempting to apply Turing's ideas on testing intelligence *in practice* (albeit the "intelligence" being tested turned out to be a hoax—a gentle practical joke at Hofstadter's expense). I consider it significant that, in operation, this turned out to be much closer to my description of the *Penrose* Test than a *Turing* Test proper. Hofstadter also, incidentally, anticipates the notion of subcognitive probing, subsequently elaborated by French. Notwithstanding this, Hofstadter's conclusion, at that time at least, was that he was (still) an “unabashed pusher of the Turing Test as a way of operationally defining what it would be for a machine to genuinely think” (p. 525).

So my final answer to French is to strictly disagree with his criticism, and insist that the Turing Test is still essentially as satisfactory as when Turing first proposed it; but I admit that the formal aspects of the Test are distracting, and I actually propose the Penrose Test as a clarification of Turing’s central idea. Indeed, while I shall continue to refer to “Turing” testing in what follows, this should now be interpreted (where this makes any difference) as *Penrose* testing.

3.3 The Problem Situation in AI

Turing’s answer to his own reformulated version of the question of machine intelligence was that he believed that a suitably programmed digital computer probably *could* pass his Test. Indeed, he went so far as to predict that “at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted” (Turing 1950, p. 442). This was perhaps somewhat rash. It is now clear that the implied target of programming a computer such that it is capable of passing the Turing Test, by the end of the century, will not be achieved; indeed, there is little consensus as to when, *if ever,* a Test might (with any confidence) be rescheduled for![5]

To be fair to Turing, he was not at all dogmatic. He explicitly stipulated that his claim (that a computer could be made to pass the Test) was conjectural and speculative; that he had no decisive arguments to show that it was possible (even in principle—never mind in practice); and that its truth could only be definitively established by exhibiting a working example of such an “intelligent” computer. Of course, it *was* an essential part of Turing’s paper to consider and

[5] Granted, the annual *Loebner Prize Competition*, launched in 1991, is derived from the idea of the Turing Test (Campbell & Fejer 1991). However, it is based on an extremely impoverished version of the Test, in that each subject can only be interrogated on a single, specified, topic, and the interrogators “were told to hold normal conversations, not to try aggressively to unmask the contestants with tricky questions” (Strok 1991). I note that the 1991 prize for the best performing computer subject (whose topic was “whimsical conversation”!) was presented, with no apparent sense of irony, not to the subject itself but to its programmer.

discount arguments *against* even the possibility of a computer passing the Test: for otherwise the formulation of the Test would have been pointless. By way of conclusion, Turing admitted that, at the time of writing, it was very unclear how best to go about trying to make a computer pass the Test, or even what the basic hardware requirements might be. Thus, Turing's achievement was in sharply defining an interesting problem, rather than offering a substantive theoretical insight into its solution. It is my view that the problem situation in Artificial Intelligence can still be quite well characterised in the way outlined by Turing. Specifically, I suggest that:

- The Turing Test has not been shown to be *invalid* (i.e. not a sufficient test for intelligence). Indeed, I would argue that this question will not actually become pressing until (or unless) some system other than a human being (whether a programmed computer, or something else, as yet unimagined) actually passes it.

- Turing Test performance has not been shown to be impossible, or inherently impractical, for a computer (not even if we restrict attention to those computers which are *already* technically feasible).

- Conversely, no essentially new argument has been forthcoming to suggest that Turing Test performance definitely *is* possible (even in principle) for a computer (either now or in the future).

- We still lack any comprehensive understanding (theory) of what, specifically, would be required to make a computer pass the test. There has, of course, been a major research effort over the 40 years since Turing's original assessment of the situation. This has yielded considerable insights into the problem. I review some of this work in subsequent sections. There is no doubt that our understanding of the *difficulties* in achieving Turing Test performance from a computer is now much more acute than when Turing first formulated the problem; but it is certainly *not* the case that we now know "in principle" how to achieve this performance, but only lack (say) adequate hardware, or a sufficient software development effort, to bring it about.

3.4 On Cognitive Architecture

Turing, through the notion of the Universal Computer, provided an *existential* argument for Artificial Intelligence: it seems that *some* (universal) computer, running *some* program, should "surely" be able to pass the Turing Test. This is the hypothesis of *Turing Test Computationalism* (H_t).

Turing Test performance would mean (by definition) that we impute "mental" states and events to such a machine. This can be done even without a commitment to the view that the machine "really" has any mentality: Dennett (1971) refers to this process as the adoption of the *intentional stance*. More generally, even for a machine which does not achieve full Turing Test performance, the behaviour may still be such as to justify a limited adoption of the intentional stance, i.e. the imputation of mentality, albeit in some impoverished form. In this way we sidestep, even if only *pro tem*, the metaphysical debate as what "genuine" (human) mentality actually consists in.[6]

Since the *un*-programmed computer manifestly lacks mentality, we thus effectively impute mentality to the computer *program(s)*: that is, the general notion that there can exist programmed computers which are intentional, or the more particular notion that there can exist programmed computers which can pass the Turing Test (H_t), implicitly asserts that the mental states and events of (or imputed to) such machines are, in principle, reducible to, or identifiable with, states and events of their programs, which is to say of purely "computational" entities.

The point is that, *whenever* we adopt the intentional stance toward a programmed computer, we implicitly identify some reduction of mental entities to computational entities.

It is the *nature* of these reductions that is actually of central interest—particularly, though not exclusively, for whatever light this might ultimately cast on *human* mentality. Admittedly, we would need to do a good deal more work to justify any step from "machine" mentality to human mentality: we would, for example, have to appeal to some *convergence* principle, to suggest that similar

[6] This is the distinction (insofar as there really is one) between H_t and the stronger doctrine of (unqualified) *Computationalism* (H_c): H_c claims that a computer which passes the Turing Test really does exhibit genuine mentality—that mentality just *is* some particular kind of computational activity.

reductions might be expected to apply to any systems, including human beings, exhibiting the relevant behaviour (Harnad 1989). However, since the level of machine "mentality" which has been achieved to date is extremely limited (falling far short of Turing Test performance) it seems a little premature to worry unduly about the ultimate scope of any such putative "machine psychology" at this stage.

Now, computational *universality* (e.g. Lewis & Papadimitriou 1981) guarantees that if a realisation of *any* abstract universal computer can pass the Turing Test (exhibit intelligence) when suitably programmed then, in principle, some (sufficiently "large" and/or "fast") realisation of *every* abstract universal computer can. Which is to say that the underlying programming formalism (which defines the abstract computer), once it is universal, does not *constrain* the intelligence of the machine. However, it *will* radically affect the *reduction* of mental to computational entities; this reduction will, at best, be unique only relative to a particular programming formalism.

Furthermore, we must at least recognise the possibility that the reduction may therefore be much *simpler* relative to some formalisms compared to others; and that which formalism is most illuminating may even be *different* depending on the particular aspects of mentality under consideration at any given time.

My point here is to distinguish between the *existential* and the *pragmatic* aspects of the Artificial Intelligence research programme. From an existential point of view, all (abstract) universal computers are equally powerful; if one can be made intelligent, they all can. But, from the pragmatic point of view, in terms of understanding or explaining intelligence (reducing the mental to the computational), there may be very substantial differences between universal computers, and, indeed, the most pragmatically useful computer may be different in different contexts.

To put it another way, consider the case of a reasonably good computer chess player—Dennett's (1978b, *passim)* prototypical example of a machine to which we can effectively adopt the intentional stance. One possible explanatory schema would be to attempt a *direct* reduction of the intentional attributions to characteristics of the program *as expressed in the native instruction set of its processor.* At best this will be unintelligible; at worst it will be hopelessly impractical. Instead, any effective explanation would certainly make use of a *hierarchy* of "levels" which *progressively* reduces (or explains) the intentional attributes. It is quite plausible that explanations at some of the different levels, or even within certain levels, may be effectively expressed in different formalisms (i.e. in terms of different *virtual machines*). It would be quite sterile to then argue about which of these formalisms is the "correct" one.

Compare also Dennett's (1978a) own discussion of a similar mode of explanation, which he describes in terms of progressively *discharging homunculi*, and Dawkins' (1986, p. 13) notion of *hierarchical reductionism*. While Dennett's discussion is most naturally applied to the decomposition of a program within a single programming formalism, similar principles would apply to transitions between different formalisms. More generally, while I have talked loosely about distinct programming "formalisms" and "virtual machines", the transitions need not always be clear cut or precise. There are hierarchies and hierarchies (compare also, the "Strange Loops" and "Tangled Hierarchies" of Hofstadter 1979).[7]

Converse arguments apply, of course, to the *synthesis* of intentional systems. While, "in principle", any abstract universal computer is as powerful as any other, in practice (i.e. in terms of the ease of designing the required functionality) some may be better than others, and a variety may be much better than any single one. The latter point still holds *even* if, ultimately, a system is *implemented* by simulating the required diverse computers on a single host. Note carefully that the distinction I am drawing here has nothing to do with the relative speed or performance of distinct, physical, computers: it is concerned purely with differences between programming formalisms, or abstract computers.

I am arguing here against a tendency to adopt an extremely over-simplified view of the computationalist thesis. Given Turing's (1950) results on universal computation, and the general notion of computationalism, there seems to be an almost overwhelming temptation to interpret H_c as positing the existence of some *specific* and *unique* (virtual) machine, or "language of thought", which, implicitly,

[7] Note incidentally that, even when one has full access to the design of an artefact such as a chess computer—i.e. when one can, in principle at least, adopt Dennett's "design" stance—the reduction of the intentional to the computational may still be a very difficult problem. For example, consider Dennett's comment on "innocently" emergent phenomena (Dennett 1977a, p. 107), or Hofstadter's discussion of "epiphenomena" (Hofstadter 1979, Chapter X).

is realised in (all) human brains, and which is a sufficient formalism for the direct explanation (reduction) of all cognitive phenomena. That is, that the architecture of cognition consists of a single significant virtual machine level, and even that this machine is of some particular class (such as, say, a `LISP`-like machine).

This kind of view can easily lead to essentially sterile argumentation about the "absolute" claims of particular abstract machines—say procedural versus declarative programming, or serial versus parallel programming, or "passive" versus "active" symbols.

More recently, this kind of argument has become an element of the on-going debate between "classical AI" and "connectionism". Consider, for example, the review of this debate presented by Fodor & Pylyshyn (1988). Their conclusion is that a connectionist "architecture", insofar as such a thing is well defined, is not a viable candidate as *the* "architecture of cognition". Now their arguments based on "combinatorial syntax and semantics" seem to me conclusive in this regard. In other words, contrary to some of the connectionist rhetoric, connectionism is not (or, at least, not necessarily) an *alternative* to classical ideas on cognitive *architecture*, but is, rather, *complementary* to it, particularly insofar as it may offer insight into ways to effectively *implement* certain aspects of classical architecture.[8] But even here there is a risk that Fodor and Pylyshyn could be (mis-?)interpreted as proposing that there does, in fact, exist *some* unique (though non-connectionist) programming formalism (abstract universal computer) which is *the* "architecture of cognition". Such an interpretation (which, I stress, may not be intended by Fodor and Pylyshyn) would be almost as bad as the position they attack: the proposal of connectionist networks as *the* "architecture of cognition".

This whole discussion is fraught with difficulty, and the possibility of misinterpretation. Thus, consider, for example, the *Physical Symbol System* hypothesis of Newell & Simon (which I will denote H_{pss}):

> A physical symbol system has the necessary and sufficient means for general intelligent action.
>
> Newell & Simon (1976, p. 41)

[8] Compare also Boden's slightly earlier review of this debate, where she considered, and ultimately rejected, the idea that connectionism might represent a Kuhnian "paradigm shift" relative to "Good Old Fashioned AI" (Boden 1988, Chapter 8).

Now Newell & Simon explicitly stipulate that a key element in their formulation of H_{pss} was the invention of `LISP` by John McCarthy, which became the prototypical example of a "symbol system", and the demonstration that such a system was "equivalent to the other universal schemes of computation". But if a *symbol system* is simply a particular *class* of (abstract) universal computer, and a *physical* symbol system is simply a realisation of a member of this class, then exactly how does H_{pss} go beyond the general computationalist hypothesis, H_c? Alternatively, if we accept the literal interpretation that H_{pss} posits that *only* a particular class of universal computers can exhibit general intelligence (i.e. the claim that a member of this class is *necessary*) then the hypothesis is simply false: by the definition of computational universality, as already discussed, if any (abstract) universal computer can exhibit intelligence then they all can.

Now, as a matter of fact, I believe that what Newell & Simon mean to claim by H_{pss} is (at least) that any intelligence system must have a "virtual machine" level which is an implementation of a symbol system (of a more or less `LISP`-like sort). This would a perfectly good qualification of H_c, i.e. a perfectly good additional hypothesis about the nature of "cognitive architecture", though it would need considerable clarification. But there again, Newell & Simon's claim may, in fact, be even stronger than this: it is simply very difficult to establish, unambiguously, what exactly they intend.

As evidence that this confusion is not merely an individual failing on my own part, consider Hofstadter's attempt at a critical evaluation of Newell & Simon's position (Hofstadter 1983); Newell dissented from this sharply in his accompanying commentary (Newell 1983), stating *inter alia* that Hofstadter was "mistaken, absolutely and unequivocally" (p. 293) in at least some of his interpretation; but Hofstadter has since repeated and reinforced his criticism, albeit with some clarification (Hofstadter 1985, Chapter 26, Post Scriptum).

There the direct argument currently rests, to the best of my knowledge. However, indirect reverberations continue. Thus, Fodor & Pylyshyn (1988, p. 59) are dismissive (verging on the sarcastic) in relation to the following particular passage from Hofstadter's original paper:

> The brain itself does not manipulate symbols; the brain is the medium in which the symbols are floating and in which they trigger each other. There is no central manipulator, no central program. There is simply a

> vast collection of "teams"—patterns of neural firings that, like teams of ants, trigger other patterns of neural firings. The symbols are not "down there" at the level of the individual firings; they are "up here" where we do our verbalization. We feel those symbols churning within ourselves in somewhat the same way we feel our stomach churning.
>
> Hofstadter (1983, p. 279)

Yet: precisely the same passage has been quoted, apparently *favourably,* by Boden (1988, p. 247).

As another example of ambivalence about the notion of an architecture of cognition, consider Fodor's book *The Language of Thought* (Fodor 1976). Reflecting the definite article used in the title, the book is dominated first by the attempt to establish that "the" language of thought exists, and then by the examination of what some of "its" properties must be.

Fodor starts off his argument with the statement that "representation presupposes a medium of representation, and there is no symbolisation without symbols ... In particular, there is no representation without an (*sic*) internal language" (p. 55). This could be interpreted simply as a variation on H_c, and, as such, is hardly objectionable; it is equivalent to the claim that a computer (generally) has a single *native* instruction set (language), in which every aspect of its behaviour can (in principle) be explicated. But: Fodor also claims that "a little prodding will show that the representational system ... must share a number of the characteristic features of real languages" (p. 31), and later he speculates specifically that "the language of thought may be very like a natural language ... It may be that the resources of the inner code are rather directly represented in the resources of the codes we use for communication" (p. 156). This kind of discussion strongly suggests that Fodor has in mind a single, unitary, language, distinct from any natural language, but not all *that* different, which is sufficient for the more or less direct reduction of mental phenomena. Now I emphasise that this simplistic view is only *suggested* by Fodor's treatment. Nowhere does he explicitly state anything to this effect; and there are at least some occasions when he appears to explicitly *reject* any such interpretation, such as the following:

> It is probably a mistake to speak of *the* system of internal representations that the organism has available for the analysis of environmental events or behavioral options. Rather, in the general case, organisms have access to a wide variety of types and levels of representation, and which one—or ones—they assign in the course of a given computation is determined by a variety

> of variables, including factors of motivation and attention and the general character of the organism's appreciation of the demand characteristics of its task.
>
> Fodor (1976, p. 157)

> ... we can see that 'the' representation that gets assigned to an utterance in a speech exchange must be a very heterogenous sort of an object. It is, in effect, the logical sum of representations drawn from a number of different sublanguages of the internal language. *It is an empirical question what, if anything, these sublanguages have in common ...*
>
> Fodor (1976, p. 159, emphasis added)

Now I agree wholeheartedly with this position; but I confess that I find it difficult to interpret the rest of the book in this light. Given such a view, it would seem that a sensible first step would be to emphasise *distinctions* between different systems of representation, rather than doing as Fodor does—which is to talk of a single system of representation (*the* language of thought) which encompasses everything.

In conclusion, I suggest that the application of the notion of universal computation in Artificial Intelligence could usefully be reversed from its usual formulation. It is usual to think of computational universality as indicating that arbitrary intentional phenomena can be "explained" in terms of a *single* mechanism—implicitly, that there is some theoretical gain ("parsimony"?) in doing so. By contrast, my view is that computational universality legitimises explanations which invoke arbitrarily complex combinations of different (computational) mechanisms, because we are guaranteed that, provided these are all effectively defined, they can all ultimately be "reduced" to a single mechanism (should there be any benefit in doing so).

That is, in considering the architecture of cognition, we need not conceive of that architecture as fundamentally identified with a particular *computer* (i.e. a particular "system" of representation, or homogenous "language of thought"); rather we may think in terms of an heterogenous *network* of (abstract) machines (homunculi) specialised for different tasks, which, in aggregation, might yield something approaching human intelligence.

This is not, of course, an original idea. For example, it is very similar to Minsky's *Society of Mind* (Minsky 1986), or Hofstadter's "soup cognition" (Hofstadter 1983). Similarly, Boden has recently emphasised the need to take a pluralistic view of computationalism, and to avoid over-simplification (Boden 1988, p. 232). Dennett has also ventured to provide a schematic design of computer "consciousness" which exemplifies these ideas (Dennett 1978c). Perhaps then I have laboured the issue unduly; and yet, I think it is clear from the literature I have reviewed above that this central point—the potential for a programmed universal computer to *simultaneously* admit descriptions both of trivial simplicity, and almost inconceivable complexity—has *not* been consistently recognised, or clearly enunciated.

The point is, in any case, crucial for my purposes here. Specifically, the work to be presented in subsequent chapters may seem, by comparison with AI research generally, to be of a relatively primitive sort; but my claim is that the problem of building, and understanding, an artificial intelligence is of unknown, but certainly vast, proportions, and that it is only with an appreciation of this that the need for the kind of fundamental research described here can be properly understood. To underline this, I close this discussion of cognitive architecture with a final quotation from Fodor:

> On the one hand, internal representations are labile and the effectiveness with which they are deployed may, in given cases, significantly determine the efficiency of mental processing. On the other hand, we know of no general constraints on how information flows in the course of the computations which determine such deployments: To say that we are dealing with a feedback system is simply to admit that factors other than the properties of the input may affect the representation that the input receives. In particular, what internal representations get assigned is sensitive to the cognitive state—for all we know, to the *whole* cognitive state—of the stimulated organism. Perhaps there are bounds to the options that organisms enjoy in this respect, but if there are no one now knows where to set them. Psychology is very hard.
>
> Fodor (1976, p. 166)

3.5 On Computational Semantics

The essence of the Turing Test, as already discussed, is to judge whether a system *means* what it says, or *knows* what is being talked about—in a comparable sense to the way we use these terms for human beings. This is considered to be diagnostic of intelligence, subsuming other aspects, such as consciousness, creativity, imagination, etc.—presumably because these latter things would, implicitly, be tested anyway: they are so central to our granting that the system understands our conversation at all that we would surely, *inter alia*, insist on testing whether it understands these particular concepts themselves.

So, the key question which arises in relation to a programmed computer which passes the Test, is: what is the relationship between its knowledge and the formal entities (tokens) making up its program?[9] Or, equally: which of these tokens mean anything at all, and, of those which do mean something, just what do they mean? Or, in the terms of the discussion of the previous section, we are asking: how are the mental entities *meaning* or *knowledge* to be reduced to the tokens constituting the computer's program?

Note again that any answer to these questions must be contingent on the definition of the computer itself: on its being the particular universal computer postulated by the program. To this extent, the reduction of meaning or knowledge to aspects of the program would not represent the complete reduction to physical terms: however, I take it that the reduction of the computer itself to physical terms will not be problematic.

We should feel much happier in attempting to answer these questions if we already had available to us a detailed specification of a programmed computer which *can* pass the Turing Test. Unfortunately, of course, we do not. Indeed, the fact is that the very design or construction of such a system presupposes, to some extent, that we already *know* what the answer to these questions is. More precisely, we must hypothesise answers to these questions as a prerequisite to building a computer system which could pass the Turing Test.

So, we need a theory of meaning or semantics, applicable to computational systems in general.

On the face of it, there is no shortage of competing theories for this purpose. I shall certainly not attempt a comprehensive review here. However, I shall suggest

[9] I use "program" without prejudice to the programming formalism, and eschewing any distinction between "instructions" and "data"; to put it another way, to the extent that any practical digital computer has finite storage, "program" can conveniently be interpreted as synonymous with the *state* (of this finite state machine). I use "token" to denote any arbitrary component of a program. It need not, and generally does not, imply any kind of "atomic" or "primitive" component of a program: indeed the entire program will be considered to constitute one particular token.

that, despite differences in vocabulary and emphasis, there is a common core to several of the theories of meaning which are in practical use within AI. I shall try to identify, and make more explicit, this common core, and to then use it as a foundation for subsequent developments.

I must consider first the view that there cannot be a "computational semantics" at all—that computers (and/or their programs) simply are not the kinds of things to which, in whole or in part, meaning can be ascribed. This position has been put forward by a number of writers. The basic idea is that a computer program is no more and no less than a formal, or syntactic object, and, as such, cannot intrinsically *refer* to anything. It may (or may not) be possible to systematically *interpret* it as referring to something, but this meaning, or understanding, is entirely in the mind of the person doing the interpretation—it is not, and cannot be, a property of the program "itself".

There is, of course, a grain of truth in this view. Thus, we may compare a computer program to a book; the book has meaning only insofar as people read and understand it. The book does not understand itself.[10] Or we may identify a program with its *information* content—in the sense of the formal theory of information; such information content would be independent of the meaning of the program (if any). Indeed, one might underline this by arguing that any particular *arbitrary* sequence of characters, of the same length as the program, would literally have the same information content (relative to an implicit, equiprobable, ensemble of all such sequences).

I said that there was a grain of truth in this view, and it is this: a program cannot be given meaning simply by wishing it so. In particular, one cannot cause a program token to mean something merely by giving it a particular label—i.e. the use of what McDermott (1976) calls "wishful mnemonics". This also applies of course *mutatis mutandis* to the tokens output by a program—as artfully, if accidentally, demonstrated by Weizenbaum with his infamous `ELIZA` program(s) (Weizenbaum 1984; see also Dreyfus & Dreyfus 1986, Chapter 3).

[10] Popper diverges somewhat from this common-sense view, with his idea of the World of *objective knowledge* (World 3). However, I do not think this is critical for the issue at hand here: while Popper would challenge the view that the only kind of knowledge is (subjective) knowledge of organisms, especially people, he certainly would not claim that a book literally understands itself.

Equally of course, the fact that a token of a program has a particular label does not necessarily mean that it is *devoid* of meaning: it merely says that *whatever* (if any) meaning it has is not *by virtue* of the label.

The *formalist* idea—that a computer program, being a purely formal object, cannot really mean anything—has been reviewed in detail, and rejected, by Boden (1988, Chapter 8). I consider that her analysis is correct, and I do not propose to repeat her detailed arguments here. The essential point is that, although a computer program *per se* may be viewed as a purely formal object, one cannot say the same for a programmed *computer*. The computer does actually *do* something (as a consequence of its program—but also depending on its inputs). This is just to repeat the point made earlier that, when one ascribes mentality to a program, this is always shorthand for referring to the system consisting of a (compatible) computer which is actually running the program. The shorthand is reasonable because we assume that the computer itself can be "easily" reduced to a lower (physical) level, should one wish to do so: the really complicated or interesting phenomena are evidently related to the particular program, rather than the computer. But while the shorthand is reasonable, it is open to misinterpretation: specifically as being a claim that a program has *intrinsic* meaning, of and in itself, *independently of any particular computer*. This latter idea is certainly mistaken, and, insofar as this is the object of the formalist's attack, the attack is justified. But the point remains that this cannot be turned into a general attack on the idea that programmed computers can have "genuine" semantics.

To return now to this main theme: following Boden, I discount the suggestion that the formal tokens making up a computer program (embedded in a suitable computer) *cannot* have "intrinsic" meaning. So far so good, but I have not yet made any positive suggestion as to what it could or should mean (!) to say that a program token *does* mean something in this intrinsic sense.

To progress this, let me first consider a terminological point. I have generally used the word *token* to refer to the constituents of programs (regardless of what the programming formalism might be), rather than the word *symbol* (except where I was citing other authors who have, or at least *seemed* to have, adopted the latter usage). My reason is, of course, that *token* doesn't prejudice the issue of semantic content, whereas once we describe something as a *symbol* we are implying that it has meaning, that it refers to something, that it serves to *symbolise*, as least with respect to *some* interpreter.

Now the conventional notion of a symbol places some emphasis on its *arbitrary* or *conventional* character: a symbol is viewed as a vehicle of communication (or, perhaps, of memory), and, as long as the communicating parties are agreed as to its meaning (even if only approximately), then the exact nature of the symbol is largely irrelevant. Of course, it is admitted that a symbol *may*, in some sense, "resemble" the thing symbolised, but this is not considered necessary or criterial for its being a symbol. This is all true and valid, but I think it may be misleading. It puts the emphasis entirely in the wrong place. While it is true that the symbol, viewed in *isolation* can be entirely unrelated to its referent (which is merely to reiterate yet again that the symbol, in isolation, *is* meaningless), the symbol, viewed in *context*, *must* be related to the referent. That is, a symbol *is* a symbol because it *is* related (somehow) to the referent.

Informally, my general claim is that what makes a token a symbol is that it interacts with some system(s) in a manner which is related, in some more or less definite (though perhaps very complex) way, to the manner in which the *referent* interacts with the same system(s). More concisely, a token is a symbol when it is used by some system to *model* something.

There is, of course, nothing novel or original in this; it is merely an attempt to spell out the implication of calling something a symbol—namely that a token is only ever a symbol *in relation to some system*, and that, further, its referent must bear, in some (identifiable) sense, a similar relationship to the system. Dennett puts it thus:

> ... nothing is intrinsically a representation of anything; something is a representation only *for* or *to* someone; any representation or system of representations thus requires at least one user or interpreter of the representation who is external to it.
>
> Dennett (1978b, Chapter 7, p. 122)

This is worth spelling out in detail because, in dealing with computer programs, it is all too easy to confound two senses in which a token can symbolise: it can symbolise something to the *programmer* and it can symbolise something to the (rest of) the *computer*. The former sense of symbolisation is, of course, the basis for the formalist argument that program tokens have no "intrinsic" meaning. The argument is wrong, but it is a very natural misunderstanding. Consider, as a more or less random example, the following quotation from Boden:

> No one with any sense would embody list-structures in a computer without providing it also with a *list-processing* facility, nor give it frames without a *slot-filling* mechanism, logical formulae without *rules of inference*, or English sentences without *parsing procedures.*
>
> Boden (1988, p. 249, original emphasis)

While Boden is here arguing for exactly the same point as I am attempting to make, it seems to me that she can easily be misunderstood. My point is that list-structures (say), in the *absence* of a list-processing facility, are only list-structures by courtesy; that is they are only list structures *relative to a human interpreter* (the programmer or otherwise). But, in the *presence* of a list-processing facility, then they become list-structures *relative to that facility.* To avoid confusion one should ideally always consistently refer to the meanings of the tokens only relative *either* to the computer *or* to a human interpreter; but, in any case, one should not switch between these two viewpoints without warning or comment, as Boden does here. The situation is not, as Boden puts it, that no one "with any sense" would embody list-structures without a corresponding list-processing facility; rather, in the sense in which Boden evidently means (i.e. relative to the computer) no one, sensible or otherwise, *could do it*—it is absolutely not a matter of choice. The idea of "embodying" list-structures in a computer without a corresponding list-processing facility is literally a contradiction in terms. The tokens in question, whatever they might symbolise to the programmer, cannot be said to symbolise (or to be) list-structures *relative to the computer* except in the case that *it* treats them so: i.e. that it *has* corresponding list-processing facilities.

Boden's version seems to be intelligible only if we interpret her to mean that, in the absence of a list-processing facility, the meaning of the relevant tokens (i.e. that they are *inter alia* list structures) should be interpreted relative to a human interpreter; *but* that, in the presence of a list-processing facility we can (and should?) change our viewpoint and interpret their meaning relative to the computer instead. But, if this *is* the interpretation Boden intends, it is poorly expressed.

Again, let me stress that this example was picked virtually at random, and I do not intend any particular or individual criticism of Boden. The problem is intrinsic to the nature of software *engineering*, and is very difficult to avoid. McDermott's idea of the "wishful mnemonic", already discussed, is obviously closely related to this.

I doubt that this point can be overemphasised. In any discussion of the "meaning" of the tokens associated with a computer program it seems that the *only* way to stay properly honest is to consistently ask "meaning relative to whom?"; and to then absolutely restrict our attributions of meaning to those which are valid or demonstrable *relative to the (programmed) computer* (or some subsystem thereof—such as some (other) specified tokens of the program). That is, we must constantly resist the temptation to, as it were, *anthropomorphize* meaning into a token.

I may add that I consider that this discipline, properly applied, seems to yield a unitary basis for a computational semantics, which resolves, or at least neutralises, the so-called *dual-calculus* view of computer programming. This view posits a sharp division of a program into active "instructions" and passive "data". Such a view leads to entirely misguided attempts to more or less exclusively or independently attribute meaning to specific aspects of instructions or data *separately.* But in fact, this distinction, while of considerable practical value in conventional software *engineering* (i.e. the development of software to exhibit particular, *effectively specified,* behaviours), lacks any intrinsic theoretical foundation.[11] That is, whether a human observer chooses to describe a particular token as "instruction" or "data", or as being "active" or "passive", or "declarative" or "procedural", is strictly irrelevant to its meaning (if any) *to the computer.* The latter meaning (which is the only meaning of interest in the AI context) can *only* be established

[11] As discussed by Hodges, in his biography of Turing, this point was already implicit in Turing's invention of the Universal Turing Machine, though it was not explicitly recognised by Turing until he set about designing his first practical digital computer (Hodges 1983, Chapter 6, pp. 324–327). It was documented in Turing's report on the *Automatic Computing Engine,* or ACE (c. 1945). As Hodges put it: "It was ... a small step to regard instructions ... as grist to the ACE's own mill. Indeed, since so much of his [Turing's] war work had depended upon indicator systems, in which instructions were deliberately disguised as data, there was no step for him to make at all. He saw as obvious what to others was a jump into confusion and illegality." Hodges also notes that this insight was not explicitly pointed out "on the American side" until 1947.

by reference to the objective interactions or effects of the token on the rest of the system (i.e. the programmed computer) in which it is embedded.[12]

In any case, to return to the question of *what* a token means (as opposed to the question of *to whom?*), we may distinguish two cases. Trivially, we may identify the meaning of the token with its *direct* effect on its interpreter. Thus we might say that a particular token "means" precisely that its interpreter should do whatever it is it does in response to that token. This notion of meaning is possible (if not very helpful) for very simple, determinate, interactions. But in more complicated cases we want to identify the meaning of the token, to its interpreter, with some aspect of the interpreter's *environment.* This is the normal usage of *symbol* or *reference*: the token *refers* (in the sense of being taken by the interpreter to refer) to some (other) thing in the interpreter's environment. And this brings us back to the notion of the symbolic token as a *model.*

This general notion of model based semantics is well established; but there is some room for debate, if not disagreement, as to what we should admit as a "model". For my purposes it is not essential to tie this down too precisely. Instead, I shall review and compare some selected ways in which it has previously been applied.

The most comprehensive, and mathematically rigorous, review of the modelling relationship of which I am aware is that of the mathematical biologist Robert Rosen (1985a). I shall therefore base my presentation of a computational semantics on Rosen's concept of an *anticipatory system*:

> An anticipatory system S_2 is one which contains a model of a system S_1 with which it interacts. This model is a predictive model; its *present* states provide information about *future* states of S_1. Further, the present state of the model causes a change of state in other subsystems of S_2; these subsystems are (a) involved in the interaction of S_2 with S_1, and (b) they do not affect (i.e. are unlinked to) the model of S_1. In general we can regard the change of state in S_2 arising from the model as an *adaptation*, or pre-adaptation, of S_2 relative to its interaction with S_1.
>
> Rosen (1985a, p. 344, original emphasis)

Where a relationship of this sort exists, I shall say that the subsystem of S_2 which is the predictive model of S_1 *means* or *refers* to S_1; and that, in this

[12] This dual calculus issue has again been thoroughly reviewed by Boden (1988, Chapter 8, pp. 248–250), though not from quite the perspective suggested here.

sense, S_2 *understands* or has *knowledge* of, S_1. I take it that, in the case of direct interest here, i.e. where S_2 is a *computational* system, then the relevant subsystem of S_2 (the predictive model of S_1) will be identifiable with a particular token of S_2's program (although this need not be the *only* function of this token, or all its components). I repeat that such "tokens" will, in general, be composite, dynamic, objects. This token then *means* or *symbolises* S_1 (to S_2); it is, genuinely or intrinsically, a *symbol.*

I do not insist that these are absolutely *necessary* conditions for meaning or grounding; but it seems to me that they may be *sufficient.*

Consider now Newell & Simon's description of what it is for a formal token to refer to, or mean something:

> A symbol structure *designates* (equivalently, *references* or *points to*) an object if there exist information processes that admit the symbol structure as input and either:
>
> (a) affect the object; or
>
> (b) produce, as output, symbol structures that depend on the object.
>
> Newell & Simon (1972, p. 21)

Condition (a) here is comparable to the requirement that S_2 (which encompasses at least the symbol structure in question and the specified information processes) must be capable (potentially, at least) of *interacting* with S_1; condition (b) is comparable to the requirement that S_2 does, in fact, contain a model of S_1. Now Newell & Simon state only these two conditions, and state them as alternatives; whereas I require both conditions, and additionally stipulate that the putatively symbolic token ("information process") must be *predictive.* Thus it is seen that my conditions for meaning or grounding are compatible with, but rather more severe than, those suggested by Newell and Simon.

I now turn to an early discussion by Dennett, in his *Content and Consciousness* (first published in 1969), where he considers the problem of the ascription of content to (physical) states and events in brains (see Dennett 1986, Chapter IV). While Dennett's concern here is mainly with human psychology, or "real" mentality, rather than with the putative mentality of a suitably programmed computer, his concepts are presented as being applicable to *any* "intentional" system, and should thus carry over more or less directly. Dennett's treatment is quite technical and involved, but is roughly summarised by the following extract:

> The content, if any, of a neural state, event or structure depends on two factors: its normal source in stimulation, and whatever *appropriate* further efferent effects it has; and to determine these factors one must make an assessment that goes beyond an extensional description of stimulation and response locomotion. The point of the first factor in content ascription, dependence on stimulus conditions, is this: unless an event is somehow related to external conditions and their effects on the sense organs, there will be no grounds for giving it any particular *reference* to objects in the world. At low enough levels of afferent activity the question of reference is answered easily enough: an event refers to (or reports on) those stimulus conditions that cause it to occur. Thus the investigators working on fibres in the optic nerves of frogs and cats are able to report that particular neurons serve to report convexivity, moving edges, or small, dark, moving objects because these neurons fire normally only if there is such a pattern on the retina. However mediated the link between receptor organ and higher events becomes, this link cannot be broken entirely, or reference is lost.
>
> The point about the link with efferent activity and eventually with behaviour is this: what an event or state 'means to' an organism also depends on what it *does* with the event or state ... Where events and states appear inappropriately linked one cannot assign content at all, and so it is possible that a great many events and states have no content, regardless of the eventual effect they have on the later development of the brain and behaviour.
>
> Dennett (1986, Chapter IV, pp. 76–77)

I suggest that this position of Dennett's is closely related to the position I have already described in relation to anticipatory systems, although Dennett provides some useful complementary insights also.

Taking first Dennett's discussion of the relationship between neural entities (or tokens, in a computational system) and "stimulus conditions", this mirrors my earlier requirement that the token be a *model* of the thing referred to. Dennett's version is somewhat more restrictive, and I would argue that it is unnecessarily so: for a subsystem might, conceivably, operate successfully as a model *without* any ongoing *linkage* to the thing modelled (i.e. without any ongoing linkage to "stimulus conditions"). However, I would grant that some such linkage to stimulus conditions may be a necessary factor in the *original* development or establishment of any modelling relationship; my point is simply that, once the modelling relationship is established, it *may* successfully persist, even if the original linkage to stimulus conditions is broken. It may also be that, given the context of Dennett's discussion, he was primarily making a *pragmatic* rather than a *theoretical* point: i.e. that only modelling relationships which (still) have extant linkages to stimulus conditions would be retrospectively identifiable in practice. Even this would be a debatable claim however. A separate point arises from my claim that the model should be *predictive*; Dennett certainly makes no explicit statement of this sort. However, his requirement that the ultimate efferent effects be *appropriate* might, conceivably, be argued as amounting to the same thing.

Moving on to the efferent effects, Dennett's general requirement that there must *be* some such effects, and that they must be *appropriate* corresponds quite well in my formulation to Rosen's stipulation that the predictive model embedded in the system S_2 must actually affect the interaction between S_2 and S_1. That is, the model must have some effect (S_2's behaviour must depend on the model), and a minimal constraint on the appropriateness of the effect is that it be concerned with the interaction with S_1. Crudely, if S_2 has a model of S_1, but only actually "uses" it in its dealings with some other (unrelated) system S_3, then we could hardly describe such usage as "appropriate". Again, Dennett's position may be somewhat more restrictive than is captured by the notion of anticipatory system: in particular, Dennett may well have in mind some more stringent tests of the appropriateness of behaviour (i.e. stronger than that the behaviour just be directed at the "right" target). I would accept that some such stronger tests might be useful in the case of biological systems, but I am not convinced that they are necessary in the general case.

In any case, it should be clear that the discrepancies, such as they are, between Dennett's view of the ascription of content and my discussion based on anticipatory systems, all rest on rather fine distinctions, and are not fundamental. In fact, I would suggest that the general notion of an anticipatory system satisfactorily captures Dennett's own idea of an intentional system, but in relatively more formal terms; that is, it seems to me that modelling a system as being anticipatory (relative to some environment in which it is embedded) is virtually synonymous with adopting the intentional stance toward it.

Next I shall consider Boden's (1988) review of the issue of computational semantics. She considers, in particular, Montague's (1974) *model-theory*:

> Model-theory deals with how meaning can be assigned to an uninterpreted formal system. Broadly, it says that such a system can be interpreted as being *about* a given domain if that domain can be systematically mapped onto it. If there are two such domains, the formal system in itself is about the one just as much as it is about the other.
>
> Boden (1988, p. 131)

It is clear that the general principle here is compatible with the view I have been describing in relation to anticipatory systems. However, it embodies only the condition that a modelling relationship must exist: it does not stipulate a *predictive* model, and, more importantly, does not require that the model must have effects on the interaction with the thing modelled. It is precisely this latter omission which introduces an unnecessary extra degree of ambiguity in the ascription of meaning. That is, if S_2 has a model which, in fact, models both S_1 and S_3, but which affects only the interaction between S_2 and S_1 (indeed, it may be that S_2 does not interact with S_3 at all), then I would argue that the model should only be said to be about S_1, and is definitely not about S_3—whereas Montague's theory would appear to admit both ascriptions equally. This is not to suggest that the anticipatory model eliminates all ambiguity of meaning: there certainly may be cases in which a model is about (affects the interaction with) more than one referent. Rather, I am saying that the conditions I propose for admitting such ambiguity are significantly more restrictive than those accepted by Montague.

It seems to me that, although this discrepancy is not too serious in itself, it may actually be symptomatic of a more fundamental difference. The fact is that, despite the suggestive overlap in vocabulary between Montague and myself, and the fact that Boden introduces Montague's work in the context of computational psychology, Montague is actually dealing with a different problem from that with which I am concerned.

My reading of Boden is that Montague is concerned with whether, in general, *isolated* formal systems can be said to mean anything. This is the problem to which his model-theory offers an answer. If I am correct in this interpretation, then I should argue that the answer is incomplete, if not actually mistaken.

This comes back to the question of *meaning to whom?* If we are really dealing with a completely isolated formal system, then I assert that it really is meaningless, regardless of what mappings may arguably exist between it and arbitrary domains. If, on the other hand, we do not really mean that the formal system is isolated, but are simply saying that it *could (potentially) be* interpreted as mapping onto certain domains (although it is not, as it were, currently or actively being so interpreted), then the theory becomes coherent but incomplete: it could only be completed by stipulating a set of (potential) interpreters.

The point is that, in the former case, if Montague is dealing with truly isolated formal systems, then his model-theory (whether right or wrong) is irrelevant—its application in my context would involve a throwback to seeing a computer program as being *purely* a formal object, a perspective I have already rejected. Conversely, in the latter case, if Montague is dealing with meaning relative to *sets* of interpreters, then, for my purposes, the theory is too general—for there is only one interpreter with which I am immediately concerned, and that is the specific system (the programmed computer) in which the formal system is embedded. With respect to *that* interpreter, I argue that the more specific (and restrictive) theory of meaning based on anticipatory systems, which specifically takes account of this particular interpreter, subsumes whatever applicability Montague's theory might otherwise have had.

Boden does not pursue this issue in depth, but it seems that her ultimate conclusions are, at least, not incompatible with my analysis. Thus, Boden ultimately merges this discussion with a more general discussion of whether a computational semantics is possible at all. There she comments that "some writers [argue] that computer programs have an intrinsic causal-semantic aspect (*to be distinguished from any abstract isomorphism there may be between programmed formalisms and actual or possible worlds*)" (Boden 1988, p. 238, emphasis added). This seems to me to be essentially the same point I have tried to make above in relation to the applicability of Montague's theory. Boden concludes:

> In a causal semantics, the meaning of a symbol (whether simple or complex) is to be sought by reference to its causal links with other phenomena. The central questions are "What causes the symbol to be built and/or activated?" and "What happens as a result of it?"
>
> Boden (1988, p. 250)

At this point, Boden's review is completed, but it should be clear that it has more or less met up with Dennett's discussion, already dealt with above. The relationship with the theory based on anticipatory systems is therefore similar, and I shall not detail it again.

I should now like to consider Harnad (1990), who introduces what he calls the *symbol grounding problem*. This is closely related to, but not identical with, my problem of the conditions under which computational tokens may be said to have meaning or reference. Harnad's problem is not identical with mine because Harnad *accepts* Searle's Chinese Room argument, and concludes from this that there are certain relatively severe constraints on symbol "grounding". I, on the other hand, reject Searle's argument (see Chapter 2 above, section 2.4.1), and therefore also reject the inferences made by Harnad from that argument. Notwithstanding this difference between us, I think it worthwhile to review Harnad's actual grounding proposals.

In this discussion, it is important to note that Harnad uses *symbol* in the special sense of a token embedded in a system of tokens which admits of "semantic interpretability" or *systematic* interpretation. That is, the meaning of any composite token (symbol) can be effectively established in terms of the meanings of its atomic components. In Harnad's terms then, the problem is that while both people and "symbol systems" can analyse the meaning of a composite token in terms of the meanings of its components, and may (via a "dictionary" or otherwise) be able to further analyse this in terms of the meanings of some (smaller) set of *primitive* atomic tokens, these primitive tokens are then, in themselves, meaningful for people but not for symbol systems. For these tokens their meaning is not (by definition) a consequence of some "definition" in terms of other tokens, so what can it be?

Harnad's outline answer is that such tokens will be meaningful or well grounded if they are derived from, or causally related to, "non-symbolic" representations of their referents, specifically in the form of what he terms *iconic* and *categorical* representations.

Iconic representations are "internal analog transforms of the projections of distal objects on our sensory surfaces" (Harnad 1990, p.342). It is quite difficult to tie this kind of idea down precisely—one immediate problem is specifying how far into the nervous system we can still speak of a “sensory surface”. But roughly speaking, Harnad means something like an *image* in the case of vision, and something analogous to this for other sensory modalities. So, an icon is something that more or less matches the sensory projection which an object would make (whenever present in the “immediate” environment) at some more or less well specified “level” or “locus” in the nervous system.

Iconic, or imagistic, representations are a well established notion in theories of mentality. I believe that it is now generally accepted that such representations (and processes constrained by them) certainly cannot account for all aspects of mentality, but it seems, equally, that they do play *some* important roles (however, see Dennett 1986, Chapter VII, and Dennett 1977b, for some critical discussion of the issue). And, of course, that is exactly the scope of Harnad’s proposal—not that iconic representations are the sole vehicles of mentality, but that they do play at least one critical role, namely being an essential step in the grounding of (primitive) *symbols* (in Harnad’s sense of that word).

A *categorical representation* of an object is then a derivative of an icon, which has been “selectively filtered to preserve only some of the features of the sensory projection: those that reliably distinguish members from nonmembers of a category” (Harnad 1990, p.342).

Harnad argues that iconic representations are necessary to allow *discrimination* of sensory inputs; that categorical representations are necessary to allow *identification* of objects (categorisation) in sensory input; and that both of these are necessary, though not sufficient, underlying processes affecting a “symbol” in order for that symbol to be well grounded. The further conditions that are sufficient for the symbol to be well grounded are that the complete system embodying the symbol must be able to “manipulate”, “describe”,[13] and “respond to descriptions” of the objects referred to. He does not go into detail of how this might be achieved, but appears to suggest that general symbol manipulation capabilities (universal computation ability operating upon the grounded symbols?)

[13] Harnad specifies that human beings can both “describe” and “produce descriptions of” objects; he clearly intends some distinction between these two, but I have been unable to understand what it is.

are the only *additional* facilities required, together with an assumption that "appropriate" processing is realised by these facilities (i.e. they have been suitably programmed?).

In terms of my own presentation of meaning in the context of anticipatory systems, these ideas of Harnad's fit in reasonably well. Iconic representations certainly are a kind of *model*, particularly suited to certain kinds of usage (namely discrimination); categorical representations are another kind of model, particularly suited to other kinds of usage (namely identification). To this extent, Harnad's proposals are compatible with mine, but are more detailed. However, Harnad appears to insist that these are *essential* components in the grounding of any symbol (whether directly, as in the case of primitive symbols, or indirectly for all others). I suggest that this condition is too strong: certainly, *some* modelling relationship is essential to grounding (to establish reference at all), but I don't see that certain particular forms of such relationship, such as those singled out by Harnad, are uniquely necessary.

More generally, I would argue that iconic and categorical representations are a very crude and limited form of model, and I suggest that they are, at best, the tip of the iceberg of general "symbol grounding".

To the extent that Harnad emphasises modelling relationships most closely related to processing of sensory input, his position is, perhaps, not dissimilar to that of Dennett. As we have seen, Dennett emphasises the need for linkage with sensory input as a basis for ascribing content. So, again, the point I wish to make is that while a close relationship to sensory input is one particularly plausible basis for establishing and/or recognising modelling relationships, it seems to me that it is not generally a *necessary* condition for the existence of such relationships.[14] I emphasise, of course, that I do not rule out the involvement of sensory input in modelling relationships; I simply stress that, in general, it may not be essential.

As to Harnad's remaining criteria for grounding (manipulation, description etc.), I suggest that these can be viewed as specific forms of my general require-

[14] Indeed, an undue reliance on sensory linkage might pose serious problems about the development of coherent mental activity despite very limited sensory ability (see Popper's remark regarding Helen Keller, quoted below). More speculatively, this is related to the persistence (at least for limited periods) of mental activity despite sensory deprivation (compare also the science fictional "brain in a vat" kind of question—e.g. Dennett 1976). However, these are extremely complex issues which I shall not attempt to discuss in detail here.

ment for the token (the predictive model) to affect the interaction with the object referred to.

As a final comment here on Harnad's discussion of symbol grounding, I note that he specifically cites Fodor (1976) as introducing semantic interpretability as criterial (or perhaps even sufficient?) for a "token" to be a "symbol". The subsequent thrust of Harnad's analysis is to reject such a purely "symbolist" view of symbol grounding, culminating in his introduction of iconic and categorical representations as alternatives to this view. Now, I have already noted, in the previous section, that Fodor could be misinterpreted in his book as proposing a unitary language of thought, at least on a superficial or cursory reading; and this seems to be precisely what Harnad has done. But, that it *is* a misinterpretation (or, at least, an oversimplification) should be clear from the fact that Fodor included, in the book, an extended and positive discussion of the use of *images* (icons?) and *discursive descriptions* of images (categorical representations?) (Fodor 1976, Chapter 4, pp. 174–194). This closely parallels Harnad's discussion, yet Harnad makes no reference to it.

To complete this sketchy review of computational semantics, I turn finally to Popper. As discussed in Chapter 2, Popper is no friend of physicalism, or, more especially, of computationalism. However, notwithstanding this, I wish to argue that Popper's *general* epistemology is compatible with the theory of meaning or reference being propounded here (contrary perhaps, to Popper's own wishes and beliefs), and can, indeed, serve to illuminate this theory further. This is a rather important point to establish, since I shall be drawing heavily on Popper's work in subsequent sections, particularly in relation to the *growth* of knowledge.

Popper's concepts of the Worlds 1, 2, and 3 have already been introduced in Chapter 2. The point argued there related to the reducibility or otherwise of World 3 to World 2 (and, in turn, of World 2 to World 1). Popper argues for their irreducibility, and in this sense, he envisages that there exists some form of "knowledge" which is *not* accessible to, or realisable by, computers. By contrast, I have claimed that Popper's argument is flawed, and thus it does not follow that what I call "computational semantics" is *necessarily* impoverished in some sense. However, I do not want to reopen that particular debate here; my immediate objective is more modest, and will be pursued separately. I wish to establish, firstly, that there is *some* sense in which Popper admits that machines, such as computers, can realise or embody “knowledge”; and secondly, that *this* kind of knowledge is essentially equivalent to the various formulations of a computational semantics which I have already discussed above.

Popper himself has not, as far as I am aware, given *detailed* consideration to the question of whether, or how, his epistemology can be applied to machines in general, or computers in particular; that is, he has not dealt very explicitly with the application of “knowledge” or “knowing” to computers and/or their programs. However, we may glean a satisfactory insight into his views by examining a variety of his writings.

There is first a discussion (dating originally from the period c. 1951–1956) in which Popper introduced the idea of a “predicting” machine, and spoke loosely in terms of its being “endowed” with, or being an “embodiment” of, knowledge (Popper 1988, pp. 68–77); granted, within that same discussion, Popper explicitly cautioned that he should not be taken as subscribing to the doctrine that “men are machines”; but, to repeat, that is not the issue just here.

Popper has consistently stressed the continuity of the biological world—that his idea of *subjective knowledge* (at least) allows for some kind of continuum, linking all living things. For example, he notes that subjective knowledge “should better be called organismic knowledge, since it consists of the dispositions of organisms” (Popper 1970b, p. 73). Now he has not explicitly used this phrase (subjective knowledge) in relation to machines, but he has, in a discussion of biological evolution, which relates specifically to the development of subjective knowledge, actually used a machine, *in place of a living organism,* to illustrate his argument (Popper 1961, pp. 274–275). Granted, Popper there emphasises that he leaves open the question of whether “organisms are machines”; but it is clear that, to some extent at least, he allows that his notion of subjective knowledge may be applicable to machines.

An underlying issue here is the (apparent) distinction between *computers* and *robots.* Thus, while Popper is dismissive in general of the “intelligence” of computers *per se,* he seems less definite about robotic systems. Specifically, the hypothetical machine referred to above, which Popper actually describes, *inter alia,* as a “complicated organism”, is, in fact, a robotic aircraft. Popper even goes so far as to refer to this machine's " 'mind' " (the inner scare quotes are, however, Popper's own).

This notion—that there is some fundamental distinction between the capabilities of computers *per se* and (computers embodied in) robots—is not uncommon. As already mentioned in section 3.2.3, French (1990) has viewed android capability as essential even to the passing of the conventional, strictly linguistic, Turing Test. For somewhat different reasons, relating to Searle's Chinese Room argument, Harnad (1989) has proposed what he calls the *Total Turing Test* which explicitly calls for full android imitation of human abilities. Boden (1988, pp. 242–245) has also taken robotic embodiment (of a computer) as a decisive element in responding to the Chinese Room argument (though her response is somewhat different from that of Harnad). A variety of other workers have, on more general grounds, advocated some form of robotic embodiment as a more or less essential aspect of realising AI (e.g. Dennett 1978d; Brooks 1986; Beer 1990; Cliff 1990).

This is a difficult issue, which I do not propose to discuss in depth here. I shall, however, state a position. It seems to me that any given system (computer or otherwise) may, potentially, be linked or coupled with its environment in an indefinitely large number of distinct ways or modalities, perhaps even with a continuum of alternatives in between these modalities. I do not doubt that the exact nature of these linkages affects the potentialities of the system. But I hold that we have, as yet, very little if any theoretical understanding of these phenomena; and, in particular, we have not yet got any basis for making a *fundamental* distinction between systems having only, say, a purely "linguistic" (VDU or teletype) interface, and systems having more extensive "human-like" or "robotic" linkages.

Returning to Popper, I may say that the apparently suggestive implication of his choice of a robotic machine rather than something closer to an isolated computer is, in any case, largely nullified by his constant rejection of "sense data" as critical for, or (worse) constitutive of, knowledge. Thus, for example, we have the following analysis:

> According to psychological sensualism or empiricism, it is the sensory input of information on which our knowledge and perhaps even our intelligence depend. This theory is in my opinion refuted by a case like that of Helen Keller whose sensory input of information—she was blind and deaf—was

> certainly far below normal, but whose intellectual powers developed marvellously from the moment she was offered the opportunity of acquiring a symbolic language.
>
> Popper & Eccles (1977, Chapter P4, p. 124)

Incidentally, Turing made much the same point at an earlier date, also citing the example of Helen Keller, and specifically applied this in the context of AI (Turing 1950, p. 456).

Thus far, I have simply argued that Popper should not be read as claiming that his theories of knowledge *cannot* be applied (at least to some extent) to programmed computers. It remains to actually apply them in this manner. I think the following quotation shows fairly clearly how this may be done:

> Because all our dispositions are in some sense adjustments to invariant or slowly changing environmental conditions, they can be described as *theory impregnated,* assuming a sufficiently wide sense of the term 'theory'. What I have in mind is that there is no observation which is not related to a set of typical situations—regularities—between which it tries to find a decision. And I think we can assert even more: *there is no sense organ in which anticipatory theories are not genetically incorporated.* The eye of a cat reacts in distinct ways to a number of typical situations for which there are mechanisms prepared and built into its structure: these correspond to the biologically most important situations between which it has to distinguish. Thus the disposition to distinguish between these situations is built into the sense organ, and with it the *theory that these, and only these, are the relevant situations for whose distinction the eye is to be used.*
>
> Popper (1970b, pp. 71–72, original emphasis)

While this quotation centers on subjective knowledge relating to sense organs, I take it that the general principles espoused here can be applied quite generally. Given this, I suggest that what Popper calls an "anticipatory theory" can be identified with what I have called a "predictive model". What he calls a "disposition" thus exists only in the context of an anticipatory theory (or predictive model), and can be identified with the contingent interaction of an anticipatory system (S_2) with the system of which it has a model (S_1). But these are precisely the conditions under which I have said that the model refers to the object modeled, or that the anticipatory system has *knowledge* of it. Thus, I claim that the knowledge which I propose to ascribe to (tokens of) computer programs is a *bona fide* case of subjective knowledge in at least one sense which Popper would allow or recognise.

While it is not crucial for my purposes here, I may say that Popper seems to distinguish this limited or impoverished kind of subjective knowledge (which, evidently, a computer *may* realise) from the more general kind (which, on Popper's view, a computer *cannot* realise) by reference to a hierarchy of *linguistic* functions. Specifically, Popper allows that both animals (including the human one) and machines can support two "lower" functions (the "expressive" and "signalling" functions), but he goes on to argue that there exist (at least) two further functions (the "descriptive" and "argumentative") which seem to be exclusively human achievements. Popper has discussed these ideas in a number of publications, but the clearest and most pertinent to the question of "machine" knowledge, is probably his paper on *Language and the Body-Mind Problem* (Popper 1953). In any case, the arguments for this hierarchy of language functions, and for the resulting cleavage of Popperian "subjective knowledge" into a kind that computers *can* realise and a kind that they *cannot* realise, seem to me to be largely equivalent to Popper's more general arguments against the causal closure of World 1. So, once again, I shall not pursue these questions further here.

This covers the application of subjective knowledge to computational systems. It leaves open the relevance, if any, of what Popper has called *objective* knowledge. Popper has generally identified objective knowledge with World 3, and I use the terms synonymously. As already noted, Popper has argued that this World 3 is not completely reducible to World 2. Still without reopening that debate, I want to emphasise that Popper accepts that in many cases World 3 objects can, in some sense, be identified with (if not reduced to?) World 2 or even World 1 objects. That is, objective knowledge can be *physically embodied.*

In general, in speaking of embodiments of World 3 objects, Popper has in mind *linguistically expressed* theories (which can then be subject to discussion and criticism). If this were the only case of the embodiment of objective knowledge, then it might have little immediate relevance to Artificial Intelligence—given the limited linguistic capabilities of existing computer systems. However, Popper also considers the concept of objective knowledge in a more general sense:

> . . . all the important things we can say about an act of knowledge consist of pointing out the third-world objects of the act—a theory or proposition—and its relation to other third-world objects . . .
>
> Popper (1970a, p. 163)

It seems to me that the implication here is that subjective *knowledge,* as such (as opposed to other World 2 objects, such as hopes and fears and pains etc.) should, quite generally, be viewed strictly in terms of its relationship to some World 3 object (the object being "grasped", in Popper's terms). In terms of my discussion of Anticipatory Systems then, I suggest that the predictive model *itself* (separately from the token which embodies or realises it) can be identified as an example of objective knowledge. That is, any systems (including computational ones) which can be ascribed subjective knowledge can also be said to *grasp,* even if only in a rudimentary fashion, some objective knowledge. They have, as it were, a toehold (at least) into World 3. This is a significant point, to the extent that Popper has stressed that, as a methodological guideline for scientific research, one should concentrate on the World 3, *objective* knowledge of a system, rather than its subjective World 2 realisation. He makes this kind of argument explicitly for non-linguistic (biological) organisms in (Popper 1968, p. 112–114).

In the present case, the significance of this is simply that, if we wish to ascribe meaning to a token (of a program) we can only do so by reference to the model which (we claim) it embodies. That is, the (tentative) identification of the World 3 object which a token embodies would be a crucial methodological step in any practical application of the computational semantics presented here.

That completes my review of computational semantics, or of the idea of artificial, computational, *knowledge.* In conclusion, then, let me just reiterate the central theme, which I have tried to view from a number of different perspectives. This is, firstly, that a programmed computer is a *dynamic* system, which must not be confused with the static, formal, object, which is its program; and secondly that such programmed computers can and should be said to be *knowledgable* precisely to the extent that they embody predictive model(s) of the reality in which they are embedded *and* that they *use* these predictions to condition their confrontation with that reality.

3.6 On the “Engineering” of Knowledge

At this point I have more or less identified the problem of Artificial Intelligence with the problem of *artificial knowledge*, and I have elaborated what I intend by that latter phrase in some detail. To be sure, for our computer to (say) pass the Turing Test, it must not only know about the world, but also be able to communicate (linguistically) about it; and to be sure, the specific problems associated with relating its knowledge to linguistic expression are far from trivial; but I suggest that the primary problem, in the current state of the AI art, is not that computers cannot talk, but rather that they have nothing worthwhile to say.

For example, as far as computer linguistic performance goes, we may consider Winograd’s `SHRDLU` system to be a climax of sorts (Winograd 1973; Hofstadter 1979, pp. 627–632). The emphasis in the development of `SHRDLU` was on language “interpretation” or “understanding”, rather than on language “production”, but perhaps this is the harder of the two. In any case, viewed purely in terms of its ability to use language, `SHRDLU` was a considerable achievement—it could indeed maintain quite a creditable and coherent conversation.

Unfortunately, the conversation turns out to be extremely monotonous, or even boring. `SHRDLU`’s “knowledge” of the world is limited to an extremely narrow and restricted domain, or a *microworld.* `SHRDLU`’s particular microworld may be thought of as a table top with various kinds of toy-like blocks on it—cubes, pyramids, etc. in various colours. There is also an arm (belonging to, or operated by, `SHRDLU`) which can be used to move these objects around. I say that the microworld may be *thought of* in this manner, but, as always, one must be careful about who (or what) is doing the thinking here. `SHRDLU`’s knowledge certainly encompasses some of the most salient aspects of the microworld I have described; but it also lacks all of the background ramifications that the description I have given would have for a human. Thus, not only is the *scope* of `SHRDLU`’s knowledge very limited, but so also is the *depth.* It might be more accurate to describe `SHRDLU`’s microworld as consisting of a ‘table top’, with ‘blocks’ ‘on’ it, of various ‘shapes’ and ‘colors’ etc.—using the scare quotes to emphasise that, although `SHRDLU` may use these terms in conversation, its understanding of them is, at best, a pale shadow of the normal human understanding of them.

So, it seems that the central problem is knowledge. It may, or may not, be a difficult problem to "hook up" an already knowledgable subject, so that it could communicate linguistically; but this problem hardly even arises until the system is quite knowledgable to start with: until it shares enough knowledge of the world with us that it might conceivably have something substantive to communicate. I emphasise this distinction between knowledge and the ability to linguistically communicate it, because there is sometimes a danger of confusing language and its content. This is closely related to the issue dealt with in the previous section, of the difference between a program *per se* (which knows nothing) and a programmed computer (which may or may not know something).

Now, the simplest conceptual approach to the problem of artificial knowledge is to *engineer* it. That is, one attempts to explicitly formulate model(s) of reality, and then instantiate them in a computer program; in other words, one builds an anticipatory system by actually designing and building the requisite predictive model(s) and using the output of these models in some (more or less) rational or appropriate way to condition the behaviour of the system—in particular, to condition its interaction with the object(s) modelled.

Knowledge Engineering is thus a brute force, or stipulative, approach to realising AI. It is the approach which has dominated AI research until relatively recently. It is (in effect) the way `SHRDLU`'s knowledge was created, and is characteristic of AI's principle commercial success, the notion of the *Expert System.*

The question which now arises is: what is the scale of this (knowledge engineering) task? How much knowledge[15] does a typical human being have? Or, perhaps slightly less demandingly, how much knowledge would be required to pass (or even come close to passing) the Turing Test?

Turing himself attempted this kind of analysis. He first estimated the "storage capacity" of the brain at about 10^9 bits, and then comments:

> At my present rate of working I produce about a thousand digits of programme a day, so that about sixty workers, working steadily through the fifty years might accomplish the job, if nothing went into the waste-paper basket. Some more expeditious method seems desirable.
>
> Turing (1950, p. 455)

[15] I pretend, for the sake of the discussion, that there could be some meaningful quantitative measure of knowledge—say something like "person-years of development effort" to realise the corresponding artificial predictive model(s).

With the benefit of forty years experience of the problems of large scale software engineering, we might be permitted some wry amusement at Turing's even contemplating the idea of developing roughly 100 MByte of software, without anything going in the "waste-paper basket"; furthermore, Turing admits that 10^9 bits is a low estimate for the storage capacity of the brain. However, the point is that these factors only serve to further strengthen Turing's conclusion that, even supposing the knowledge engineering approach to be *theoretically* tractable, it is not *practical.*

While we might now be more reticent about doing this kind of calculation, nothing in the past forty years has served to suggest that Turing may have seriously *under*-estimated the effort required. That is, the knowledge engineering approach has proved more or less successful in narrow domains of knowledge, but it has remained limited to such domains. In terms of the original objective of general intelligence, at the Turing Test level, the approach has largely stagnated.[16]

The apparent limitations of knowledge engineering have been recently documented by Hubert and Stuart Dreyfus (Dreyfus & Dreyfus 1986). They identify two related difficulties: the *common sense knowledge* problem, and the *frame* problem.

The common sense knowledge problem refers to the extreme difficulty which has been encountered in attempts to systematise common sense knowledge. This is generally agreed to be a very severe problem, though there is room for debate as to its exact nature—specifically, whether it is "merely" a matter of scale, of the sheer quantity of knowledge involved, or whether there are more fundamental problems not yet properly recognised. Thus, for example, Hayes (1979) proposed a research programme to systematise or formalise "a large part of ordinary everyday knowledge of the physical world"—what he dubbed *naïve physics.* Drew McDermott was originally an enthusiastic advocate of Hayes' approach, but subsequently (McDermott 1987) reported that very little progress had been made, and concluded that the programme faced very fundamental and substantial difficulties.

[16] One major exception is Lenat's `Cyc` project (Lenat & Guha 1990). However, substantive results (one way or the other) are not expected from this project before about 1994.

The frame problem refers to the fact that, even if a system has been provided with a great deal of knowledge (without, for the moment, trying to quantify this), it is very difficult to integrate this successfully—especially to ensure that the most relevant knowledge is available and applied at any given time; and, of course, the more knowledge is provided, the worse this problem becomes. Dreyfus & Dreyfus describe the frame problem as follows:

> In general skilled human beings have in the areas of their expertise an understanding that enables them, as events unfold, to distinguish what is relevant from what is not. However, during the first three phases of AI research, from cognitive simulation up through work on micro-worlds, computers, like beginners, advanced beginners, and competent performers, were programmed to confront all facts as isolated from each other and goals as just further facts. Thus whenever a change occurred the whole set of facts that made up the computer's representation of the current state of affairs had to be recalculated to update what had changed and what had remained the same. The attempt to capture human, temporal, situated, continuously changing know-how in a computer as static, desituated, discrete, knowing that has become known as the frame problem.
>
> Dreyfus & Dreyfus (1986, p. 82)

It is interesting to compare this with Popper:

> At every instant of our pre-scientific or scientific development we are living in the centre of what I usually call a *'horizon of expectations'.* By this I mean the sum total of our expectations, whether these are subconscious or conscious, or perhaps even explicitly stated in some language. Animals and babies have also their various and different horizons of expectations though no doubt on a lower level of consciousness than, say, a scientist whose horizon of expectations consists to a considerable extent of linguistically formulated theories or hypotheses.
>
> Popper (1949, p. 345)

My point here is that while Popper has never explicitly addressed the frame problem, it is clear that his theory of knowledge encompasses the issues it raises. To the extent that I have argued in the previous section that computers can, in principle at least, realise knowledge in Popper's sense, this can be taken as a claim that computers can, in principle, be programmed to overcome the frame problem. This is worth stating explicitly because Dreyfus & Dreyfus are frankly skeptical about it.

However, the point remains that, to date, the brute force method of knowledge engineering has proven to be extremely limited as an avenue toward the realisation of artificial intelligence. It seems that some alternative should be sought.

3.7 Building a Baby

Turing himself had, of course, anticipated that what I now call the knowledge engineering approach might prove impractical. His proposed alternative was to develop a machine which would be capable of *learning.* In this way, he hoped, the *initial* programming requirement could be reduced to manageable proportions:

> Presumably the child-brain is something like a note-book as one buys it from the stationers. Rather little mechanism and lots of blank sheets. (Mechanism and writing are from our point of view almost synonymous.) Our hope is that there is so little mechanism in the child-brain that something like it can be easily programmed. The amount of work in the education we assume, as a first approximation, to be much the same as for the human child.
>
> Turing (1950, p. 456)

This possibility has certainly not been ignored in the intervening years. However, Turing's "hope" that it might prove significantly easier to program a "child-brain" compared to an adult has, so far at least, proved forlorn. As Charniak and McDermott put it:

> One idea that has fascinated the Western mind is that there is a general purpose learning mechanism that accounts for almost all of the state of an adult human being. According to this idea, people are born knowing very little, and absorb almost everything by way of this general learner. (Even a concept like "physical object," it has been proposed, is acquired by noticing that certain visual and tactile sensations come in stable bundles.) This idea is still powerfully attractive. It underlies much of behavioristic psychology. AI students often rediscover it, and propose to dispense with the study of reasoning and problem solving, and instead build a baby and let it just learn these things.
>
> We believe this idea is dead, killed off by research in AI (and linguistics, and other branches of "cognitive science"). What this research has revealed is that for an organism to learn anything, it must already know a lot. Learning begins with organized knowledge, which grows and becomes better organized. Without strong clues to what is to be learned, nothing will get learned.
>
> Charniak & McDermott (1985, pp. 609–610)

Popper has made related claims in a different context:

> ... to every man who has any feeling for biology it must be clear that most of our dispositions are inborn, either in the sense that we are born with them (for example, the dispositions to breathe, to swallow, and so on)

> or in the sense that in the process of maturation, the development of the disposition is elicited by the environment (for example, the disposition to learn a language).
>
> Popper (1970b, p. 66)

> If it were not absurd to make any estimate, I should say that 999 units out of 1,000 of the knowledge of an organism are inherited or inborn, and that one unit only consists of the modifications of this inborn knowledge...
>
> Popper (1970b, p. 71)

Boden (1988, p. 187–188) also discounts Turing's original programme for similar reasons. Somewhat more caustically, but making an essentially similar point, the biologists Reeke & Edelman have said:

> In fact, consideration of the magnitude of the problem with due modesty suggests that perception alone is hard enough to understand, without attempting to jump directly from perception to learning, through learning to social transmission and language, and from there to all the richness of ethology. At present, it is still a large challenge to understand how an animal can even move, and it would be well for AI to look first to such fundamental issues.
>
> Reeke & Edelman (1988, p. 144)

I note that this criticism by Reeke & Edelman is directed just as much at the recent revival of research in the field of artificial neural networks ("connectionism") as at the approaches of conventional (so called "symbolic") AI.

This result—that the realisation of an artificial "infant" intelligence seems not to be significantly easier than the realisation of an "adult" intelligence—is certainly disappointing, but it is by no means completely fatal to the AI enterprise. With my rejection, in the previous section, of the knowledge engineering approach the problem had already changed from that of artificial knowledge in itself, to the problem of the *growth* of (artificial) knowledge. That still remains our problem, but now we recognise that this cannot be solved by restricting our attention to the somatic time growth of human intelligence. We must take a more comprehensive view, in which human knowledge is seen as being continuous with animal or biological knowledge. In short, we must understand not only "learning" but also "evolution".

3.8 The Growth of Knowledge

The development so far has indicated that, in attempting to realise artificial intelligence, we must realise the *growth* of artificial knowledge; and that, furthermore, we must be concerned not just with the somatic time "learning" of an individual, but also with the evolutionary time growth of "inate" knowledge. That is, we must ultimately seek to realise the growth of artificial knowledge in something comparable to *both* of these biological senses.

I now wish to go beyond this result, and review a stronger, perhaps even a radical, claim: this is that the growth of knowledge by learning and by evolution are not fundamentally distinct processes in any case—rather, they are both forms of a kind of abstract or generalised Darwinian process. This being so, it will follow that the AI research programme may be identified with, or even replaced by, a programme aimed at the realisation of *Artificial Darwinism.*

3.8.1 Evolutionary Epistemology

The doctrine that the processes underlying *all* growth of knowledge are of an essentially Darwinian kind is now called *evolutionary epistemology*; the concept was pioneered by Popper, and it is fundamental to his overall philosophy, but it has also been significantly expanded and developed by others—Radnitzky & Bartley (1987) provide a comprehensive survey. I have very little to add to this existing literature, so I shall restrict myself here to a relatively brief review.

Evolutionary epistemology derives, ultimately, from Popper's analysis of the problem of *induction,* and the implications he draws from this for the growth of knowledge (e.g. Popper 1971). Popper denies that there can be such a thing as *certain* knowledge (except in the trivial sense of a tautology); and, more importantly, denies that there can be such a thing as a *logic* of induction. That is: nothing that we know is *necessarily* true (including "observation statements", since these are, themselves, theory impregnated); and even to the extent that what we know is, in fact, true, we cannot *logically* infer from it (consciously or otherwise) any more general, or strictly new, knowledge.

To take a favoured example of Popper's (e.g. Popper 1970b, p. 97) we know that the Sun rises each day; but this is not certain knowledge (there are any number of reasons why the Sun may *not* in fact rise tomorrow); and our (tentative or conjectural) knowledge that the Sun will, in fact, rise tomorrow, is not, and cannot be, a consequence of, nor justified by, our experiences of the Sun rising on previous days—no matter (for example) how many times this experience may have been repeated.

It is important to emphasise here that, in this Popperian view, deduction as such can never result in *new* knowledge or in the growth of knowledge. Deduction is a tool which can be used (and most notably *is* used in science) to draw out consequences of our existing knowledge; but this is (just) making explicit what was already implicit, and, in itself, cannot increase our knowledge. Processes weaker than logical deduction (e.g. so called fuzzy logic) cannot, of course, reliably do any better in this particular respect.

Popper has called the naïve empiricist idea that knowledge is "derived" or "extracted" or "distilled" from some accumulation of "experience" the *bucket theory of knowledge* (Popper 1949). His central point is that, as long as knowledge is interpreted in the sense of effective predictive models, the bucket theory is untenable *on purely logical grounds.*

But if knowledge cannot grow through the analysis of experience, then how *does* it grow?[17]

Popper's answer is that knowledge grows, and *can* only grow, by a process of *conjecture and refutation.* By "conjecture" he means the formulation, *by any means*, of new models or assertions or theories, which make predictions about the environment in which the knowledge agent is embedded. The only constraint is that the predictions of these new models must potentially go beyond (or conflict with) predictions made by the prior knowledge of the agent.

In effect, we must divide knowledge processes into two kinds. In the first, truth conditions (or more strictly, *belief* conditions) are preserved. Such processes clearly do not involve any growth of knowledge, but rather represent an *elaboration* of existing knowledge; they include, for example, the operation or execution of "predictive models", as I have used the term (following Rosen 1985a)

[17] I discount here the relativist (non-)answer that the growth of knowledge is an illusion; whatever about the growth of human knowledge, the idea that the growth of biological knowledge, in the evolutionary sense, is imaginary, seems to me to be quite unsustainable.

in section 3.5 above. These processes correspond to the *application* of knowledge which has been accepted or adopted (at least tentatively) by an agent. In the second kind of knowledge process, truth (belief) conditions are not strictly preserved. Since these processes are not truth preserving, their output is inherently conjectural (i.e. even relative to premisses which are assumed to be true). Regardless of the nature of these processes, we may say that they represent *unjustified* variation. Such unjustified variation is clearly distinct from elaboration of already accepted knowledge—for it potentially produces conjectures transcending, and especially *contradicting,* the previously (presumed) known. But unjustified variation does not yet represent *growth* of knowledge, for these new conjectures may be uniformly mistaken.

Knowledge can *grow* if and only if the agent's predictions ("deductions") are *frustrated*; that is, if the world does *not* behave in the way the model (the knowledge) predicts; whenever expectations are defied in this way, there is an *opportunity* for growth. But this opportunity can be exploited only if the failure has the effect of selecting between competing models which were not all equally bad at predicting the behaviour of the world.

In short, there must be *some* mechanism for generating new candidate models of reality, which may compete with, and improve upon, the old ones; these may, or may not, be derived in some sense from existing models; but their validity or utility, is independent of their genesis. Knowledge can thus continue to grow only to the extent that new models of aspects of the world, not logically (deductively) entailed by prior knowledge, can be generated and tested. There can be no definite method (or logic of induction) for the generation of such new models which would guarantee their truth. For that matter, there cannot even be a definite method for *testing* of competing conjectures.[18]

While Popper originally formulated this theory of the growth of knowledge by conjecture and refutation in the context of *scientific knowledge*, the schema

[18] I may say that I fully accept that there is a concept being used implicitly here, of relative "closeness" to the truth, or "verisimilitude", which is a difficult and problematic one; but I think it has, nonetheless, a clear and useful intuitive meaning—in this, I follow Newton-Smith (1981), at least partially (compare also Popper 1974b, pp. 1011–1012). In any case, it must be emphasised that the notion in question is always a relative one: we are always talking about a comparison between competing conjectures rather than between an isolated conjecture and the "naked" truth (the latter being strictly inaccessible—to us just as much as to our machines).

clearly represents a kind of generalised or abstract *Darwinian* process. Campbell (1974a, p. 49) points out that this Darwinian undertone can be found even in Popper's earliest discussions of the subject; in any case, Popper himself has since (e.g. Popper 1961) explicitly emphasised the essential unity of all processes of knowledge growth in both evolutionary and somatic time (and also, indeed, in what we might call "cultural time", in the case of linguistically formulated World 3 knowledge; but that further extension is not relevant to my purposes here).

Thus, under the doctrine of evolutionary epistemology, all knowledge (subjective or objective, conscious or unconscious) consists in *tentative* hypotheses or theories about the world (including theories about theories, and how to apply them etc.). Growth of knowledge is possible (indeed, is *only* possible) through the unjustified formulation of new, tentative, theories, and some form of testing and selection between competing theories. In the special case that the theories are linguistically formulated, are falsifiable, and selection is based on rational analysis and critical testing, then Popper identifies this as *scientific* knowledge. In the special case that the theories are inate or inborn, and selection is based on differential reproduction arising through competition for limited resources, then the process is conventional Darwinian evolution. But, in all cases, the growth of knowledge involves an initial unjustified generation of new conjectures—i.e. conjectures whose truth is not logically entailed by the (tentatively accepted) truth of previous knowledge—followed by a confrontation between these new conjectures and the objective world, a confrontation in which the (more) false are rejected.

Campbell (1974b) has referred to this unified theory of knowledge growth as *Unjustified Variation and Selective Retention*, or, as I shall say, *UVSR*.

3.8.2 On "Random" Variation

The variation which underlies Darwinian (UVSR) evolutionary processes is commonly referred to as being "random", but it turns out that this has connotations which can be deeply misleading. Campbell (1974a; 1974b) has previously reviewed this question quite comprehensively; I shall simply extract some details which will be particularly relevant to my own objectives here.

Unjustified variation is an essentially *logical* notion. While it perhaps conforms to one of the common-sense ideas of “randomness”, it is certainly *not* random in the sense of the probability calculus. This should be clear, for example, from the fact that the probability calculus relies on the possibility of a defined *event space*, whereas to classify a variation as “unjustified” does not *require* any reference to a “space” of “possible” variation. Lack of justification is, rather, a logical relationship between new (tentative) knowledge and prior knowledge.

More generally, “randomness” implies the absence of *predictability,* (except, perhaps, in a statistical sense) whereas “unjustified” variation may be arbitrarily systematic and predictable. Note that this does *not* imply that the growth of knowledge is predictable. As Popper (1988, pp. 62–67), for example, has pointed out, any claim to be able to predict the future growth of knowledge is fundamentally flawed; in the case at hand, such a claim must fail because the growth of knowledge requires both the generation of unjustified variations (and we stipulate that this *may* be predictable, in isolation) but also the testing, and selective retention, of some of the generated variations. This second step, of selective retention, is *not* predictable.

This establishes that unjustified variation is not (necessarily) a “random” process in the sense of the probability calculus, and that it need not even be unpredictable. But the crucial distinction between the notions of “unjustified” and “random” variation is rather more subtle than this. A key connotation of “random” variation, in the context of knowledge processes at least, is that it is “unbiased” with respect to the *truth*, or, more generally, the *verisimilitude*, of the generated conjectures. That is, a “random” variation should be “just as likely” to be *false* as to be *true.*

I should point out that, while this notion of randomness, in the sense of a lack of bias between truth and falsity, seems to have a fairly clear *intuitive* meaning, it can hardly be made formally respectable at all. To say that a process of generating conjectures is “random” in *this* sense requires that we be able to categorise all possible conjectures which it can generate as true or false *in advance*; and if we could do *that* then we would already have all the accessible knowledge in our possession, and there could be no growth. Granted, we could possibly apply this notion to our machines, or perhaps even to animals—if the domain of their knowledge is strictly circumscribed to lie within some domain of which *we* already have "perfect" knowledge. But of course, we never have "perfect" knowledge of any domain, so even this is a contrived case; furthermore, machines whose knowledge cannot, *in principle*, transcend our own would still represent a critically impoverished kind of artificial intelligence.

In any case, the important point is that the idea of "unjustified" variation does *not* require or imply "randomness" in this last sense either. A particular process for generating new conjectures may, in fact, be strongly "biased" (either towards truth or falsity), insofar as this idea of bias can be given a clear meaning at all; but we can never *know* this to be the case. Our labelling of a generation process as *unjustified* does not rely one way or the other on such bias, or its absence.[19]

In fact, it seems that, if anything, "successful" UVSR processes typically involve generators which *are* strongly biased in favour of true (or "approximately true") conjectures. The crucial point here is that, while the *possibility* of knowledge growth does not rest on such bias, the *rate* of growth will be strongly affected by it.

In saying this it may appear that I am now begging the question at issue: it seems that I now explain or solve the problem of the growth of knowledge (at least insofar as it occurs at "speed") by calling upon some mysterious inbuilt bias in the generation of new conjectures. But this is actually a flawed criticism—because the UVSR principle can be applied recursively. If I should say (or conjecture) that a generator of (unjustified) conjectures is favourably biased, I would be implying that it already incorporates knowledge; while this would raise the question of where *this* knowledge came from, I have a simple answer ready—namely that it came from a prior process of (unjustified) generation of different *generators,* with selective retention of the "best". The implied regress is *not* vicious: it can bottom out with generators which are unbiased (or even unfavourably biased)—or, more to the point, generators whose operation can be explained without any

[19] I should caution that I myself have previously made the terminological blunder of using "unjustified variation" to refer not just to the relatively weak logical concept for which I use the term here, but *also* to refer to the stronger concept which I here call "unbiased variation" (McMullin 1992a; 1992b); this was a blunder insofar as I acquired the phrase "unjustified variation" from Campbell (1974b), and it has become clear to me that Campbell meant the term to imply *only* the weak, strictly logical, notion. I have therefore now reverted to using it *only* in this original sense of Campbell's.

assumption of bias (i.e. non-teleologically) one way or another.

So the "unjustified" in UVSR does not equal "unbiased" (or "ignorant"); but a complete evolutionary epistemology *does* demand that, to whatever extent an unjustified generator is held to already incorporate significant knowledge, that the genesis of *this* knowledge must be explained by a prior UVSR process. This recursion must ultimately terminate in "primordial" generators whose genesis is not problematic. We can hold that such a termination is *possible*, even if we have little or no idea of its detailed nature, because UVSR *can* operate even in the face of a strongly *un*-favourable bias (provided enough time is allowed).

This is an important result. The previous section established, on strictly logical grounds, that *unjustified* variation was necessary to the growth of knowledge—but said nothing about the *rate* of such growth. I am now suggesting that the rate of growth will depend on an ability to exploit previously gained knowledge in a loosely hierarchical fashion: that, in other words, for "fast" knowledge growth, we need an architecture which is not tied into a fixed, predefined, generator of unjustified variation, but which instead supports the emergence of new generators of unjustified variation, and selection between such generators.

There is a further separate, but complementary, result, regarding the "openness" of knowledge growth, but it will require some discussion to develop it properly.

Note firstly that, while it seems that an organisation supporting such a hierarchic knowledge structure may be a *necessary* condition for "fast" knowledge growth, it cannot, of course, be a *sufficient* condition. We may say that the fundamental principle of evolutionary epistemology is that there are *no* sufficient conditions for the growth of knowledge.

In practical terms, this means that if we happen to find a "good" generator of new (tentative) knowledge, then that can allow a burst of relatively rapid knowledge growth, but this will inevitably be exhausted; further growth of knowledge will then rely on generating an alternative generator. That is to say, a "good" generator is a double-edged sword: to the extent that it does generate good conjectures, it accelerates the growth of knowledge; but buried among all the "bad" conjectures which it does not generate, there may be some jewels, better even than the "best" conjectures which it does generate.

Thus, once it is accepted that *all* knowledge is conjectural—including that incorporated in our best "generators"—we see that the growth of knowledge may ultimately cease altogether if we cling dogmatically to *any* knowledge. Conversely, if we wish the growth of knowledge to be (as far as possible) open-ended, we need a knowledge structure which is *not* a simple hierarchy, but is rather more like a *heterarchy*, in which *all* knowledge (including all generators) is potentially subject to competition and displacement.[20] There is this inherent tension between the two aspects of the UVSR process—between variation and retention—and it is precisely the maintenance of this tension which permits (but cannot compel) continued, open-ended, growth of knowledge.

Our problem then is to find a way to design our putative machine or computer "intelligence" in just such a way that it can successfully balance on this same knife-edge which separates dogmatism from ignorance. This is not a trivial task.

3.8.3 UVSR and AI

The notion of realising some kind of more or less Darwinian process, in a computational system, is not at all original. Turing (1950) explicitly drew parallels between processes of learning and of biological evolution, in his seminal discussion of the prospects for realising AI. The earliest practical research was probably that of Friedberg and his colleagues in the field of "automatic programming" (Friedberg 1958; Friedberg *et al.* 1959). However, the problems they tackled were extremely simple (e.g. 1-bit binary addition), and the results mediocre. A form of artificial Darwinian evolution was proposed by Selfridge (1959), but was not pursued to an implementation. Simon (1969) provided an early, and perceptive, analysis of some of the general factors involved in applying any kind of Darwinian process in artificial systems.

The idea of artificial evolution was taken up again by Fogel *et al.* (1966), now specifically in the context of AI, but still applied to very simple problems and with very little tangible success. The scathing review of this particular work by Lindsay (1968) was, perhaps, instrumental in discouraging further investigation

[20] In its most abstract form this becomes, in effect, the principle of *pancritical rationalism* advocated by Bartley (1987). Compare also what I have previously called the "reflexive hypothesis" (McMullin 1990).

along these lines for some time. Holland has since rehabilitated the evolutionary approach somewhat with the so-called *Genetic Algorithm* or GA (Holland 1975), and this has generated a significant level of interest, particularly in the USA (Schaffer & Greffenstette 1988). I shall discuss the philosophical background to Holland's work in more detail below, and will also consider the Genetic Algorithm again in Chapter 4, section 4.3.2. There has been a somewhat parallel European development in the form of *Evolution Strategies* (Schwefel 1979; 1988). Several of these historical developments have been recently reviewed by Goldberg (1989).

I shall not attempt a comprehensive discussion of these prior efforts here. They have met, at best, with limited success, and then only in relatively narrow domains. I suggest that this failure can be traced, to a large extent, to the fact that these approaches have *not* been informed by the detailed philosophical arguments and analyses which have been elaborated by Popper and others under the rubric of *evolutionary epistemology.* I shall develop this claim by considering a number of attempts to probe the philosophical foundations of evolutionary or Darwinian growth of knowledge which have previously appeared in the AI literature.

I start with the analysis contained in Daniel Dennett's paper *Why the Law of Effect Will Not Go Away* (Dennett 1975). This is a remarkable paper in that Dennett succeeds in (re-)developing many of the important ideas present in evolutionary epistemology, but appears to have done so almost independently of, and concurrently with, the "mainstream" work in the field. There is only fleeting mention of Popper, with no detailed citation; and there is no mention at all of D.T. Campbell, or other workers associated with the development of evolutionary epistemology.

I note that Dennett's paper is clearly a development of ideas previously mooted in his *Content and Consciousness* (Dennett 1986), which was originally published in 1969. Thus this work by Dennett either predated or overlapped with the original publication of Popper's collection of essays *Objective Knowledge* in 1972 (Popper 1979), and the publication of the Schilpp volume on Popper's philosophy (Schilpp 1974), which also contained Campbell's most comprehensive expression of evolutionary epistemology (Campbell 1974a). So, notwithstanding the fact that the essential ideas of evolutionary epistemology were available in much earlier publications[21] it is perhaps not too surprising that Dennett's development of these ideas was quite independent. While Dennett has mentioned Campbell's work more recently (Dennett 1981, p. 59), this was in a quite different context, and does not bear directly on the issues to be discussed here. As far as I am aware, neither Dennett himself, nor any other commentator, has previously drawn attention to the close connections between Dennett's analysis and evolutionary epistemology *per se*.

Although Dennett's treatment does not, in my view, go significantly beyond the analyses of Popper and Campbell, it is particularly relevant here because Dennett explicitly relates these ideas to AI.

Dennett expresses his discussion in terms of the thesis known, in behaviourist psychology, as the *Law of Effect*; very roughly, this states that actions followed by reward are likely to be repeated, or, more specifically, that rewards act to reinforce or *select* the "successful" behaviours from a diverse repertoire of possible behaviours. While Dennett is no friend of behaviourism, his claim is that there is a core of truth in the Law of Effect, and, in particular, that something like it will be "not just part of *a* possible explanation of behavior, but of *any* possible explanation of behavior" (Dennett 1975, p. 72, original emphasis).

Dennett develops this claim by first noting that Darwinian evolution provides an explanation of the growth of what I have called "inate knowledge" (and which Dennett refers to as purely "tropistic" or "instinctual" control of behavior); and that, for the time being at least, Darwinism is the *only* account we have of this growth which is not "question-begging". In Dennett's view, the Law of Effect provides a similarly unavoidable basis for any satisfactory account of the growth of knowledge in somatic time (i.e. what I termed "learning" above, as opposed to "evolution")—indeed he now reformulates the Law of Effect in a generalised form of what he calls "generate-and-test" procedures. These are hardly distinguishable from Campbell's UVSR processes—except that Dennett does not explicitly

[21] As already noted, there were clear anticipations of evolutionary epistemology in Popper's *Logik der Forschung*, first published in 1934, of which the English translation, *The Logic of Scientific Discovery*, first appeared in 1959 (Popper 1980); there were also substantive specific discussions of the problem of induction in Popper's *Conjectures and Refutations* (Popper 1989), first published in 1963; and Campbell published two seminal papers in the field at an early date (Campbell 1960a; 1960b).

formulate the notion of logically *unjustified* variation as a essential element of these processes.

Dennett's analysis closely mirrors that of Campbell in other respects also. In particular, he introduces the notion of an "inner" environment which can allow a selection process to proceed internally to an organism, apart from overt external behaviour. This is very similar to Campbell's idea of "vicarious" or "substitute" selectors (Campbell 1974a), and, like Dennett, Campbell has explicitly viewed this as a necessary generalisation of earlier learning theories, behaviourist and otherwise (Campbell 1960a). Following Simon (1969, Chapter 4, pp. 95–97), Dennett's discussion of the rôle of generate-and-test procedures in AI focuses on the requirement (if the growth of knowledge is to be "efficient") for generation processes to be "endowed with a high degree of selectivity" (Dennett 1975, p. 86)—which I take to be equivalent to what I have earlier called "favourable bias". However, unlike Simon, but exactly matching Campbell's various descriptions, Dennett explicitly recognises that any such "selectivity" in a generation process itself demands an explanation in turn, and that this can only be satisfied by a recursive appeal to some earlier (perhaps properly Darwinian or evolutionary) generate-and-test process; and, further, that this recursion can only bottom out with an appeal to some "ultimate" generators which "contain an element of randomness or arbitrariness" (Dennett 1975, pp. 86–87).

So: it seems clear that Dennett's ideas, so far at least, are essentially at one with the ideas of evolutionary epistemology. But I have not yet dealt with Dennett's substantive claim: that his generalised Darwinism, in the form of generate-and-test, is a *necessary* component of any satisfactory psychology (or, indeed, AI). Frankly, I find Dennett's argument here obscure, and I shall not attempt to reproduce it. But I note that, in introducing his argument, Dennett says:

> I suspect this argument could be made to appear more rigorous (while also, perhaps, being revealed to be entirely unoriginal) by recasting it into the technical vocabulary of some version of "information theory" or "theory of self-organizing systems". I would be interested to learn that this was so, but am content to let the argument, which is as intuitive as it is sketchy, rest on its own merits in the meantime.
>
> Dennett (1975, p. 84)

Now I admit that I may not have grasped Dennett's argument correctly; but insofar as I think I do understand it, it seems to me that it is, indeed, "entirely unoriginal" (as Dennett anticipated) and that it already *had* been made more "rigorous". However, far from being related to "information theory", I think it is actually, in essence, a specialised form of Popper's argument in relation to the impossibility of a logic of induction. It differs primarily in that Dennett has failed to locate the problem as precisely or clearly as Popper, as being that of the growth of knowledge (in the face of the impossibility of induction), and his solution is, as a result, much less clearcut than Popper's; but it *is* essentially the same result, namely that there is no logical alternative to seeing knowledge as irredeemably hypothetical, and that knowledge grows, if at all, by (unjustified) variation and selective retention.

Having accepted that Dennett's analysis of the growth of knowledge is essentially equivalent to the doctrine of evolutionary epistemology, the remaining question is the relevance of this to AI.

Dennett notes that generate-and-test, in the most general sense, is a "ubiquitous" strategy within AI programs (and actually uses this fact to bolster somewhat his argument for the *necessity* of such processes). That is fair enough, as far as it goes, but it does not go very far. The important point for my purposes is that the use of generate-and-test is not, *in itself,* any kind of panacea. Dennett argues (and I agree) that such processes will be necessary in the realisation of AI—but they are not sufficient by any means. I have argued earlier that the effective and open-ended growth of knowledge actually requires that the architecture of the knowledge agent must support the growth and elaboration of a heterarchical structure in which no knowledge (including knowledge generators, and vicarious selectors) is sacrosanct, or dogmatically held. This is a much stronger requirement than simply insisting on having "some" form of UVSR; as far as I am aware, such an architecture is unknown in any functioning AI system.

In particular, Dennett himself (who is, of course, primarily a philosopher) has not attempted to actually apply the abstract ideas reviewed here in the design of any real AI system, and I shall therefore not discuss his analysis any further.

I now turn to some writers who, at first sight at least, might seem to oppose the central claim that UVSR processes are essential to the growth of knowledge, and thus to the realisation of AI.

Consider first the criticism by Boden (1984). Boden is primarily concerned with the relevance of Darwinian processes to problems of mentality, specifically including AI, but she is also prepared to carry her attack to the home ground of Darwinian theory, biological evolution itself:

> Perhaps similar considerations concerning creative exploration might illuminate various biological phenomena which, on a neo-Darwinist account of evolution, are very puzzling. These include the facts that the fraction of DNA that does not code for the synthesis of specific proteins increases phylogenetically; that species have evolved remarkably quickly, and that the more complex species have if anything evolved at a greater rate than their predecessors; and that the speed with which a species evolves morphologically seems quite unrelated to the rate at which its individual proteins evolve (so frogs have protein-synthesizing mechanisms of comparable complexity to those of man). Such facts are not explicable in terms of "Random-Generate-and-Test," the mutational strategy favoured by neo-Darwinism. This is because (as was discovered by the early workers in automatic programming), the combinatorics of such a process are horrendous (cf. [Arbib 1969a[22]]). Switching to a higher-level biological language (cf. "consolidation"), might be effected by random processes of gene duplication and recombination; but this merely reduces the exponent without preventing an exponential explosion.
>
> Instead, some strategy of "Plausible-Generate-and-Test" is needed, whereby *mutations of a type likely to be adaptive become increasingly probable.*
>
> Boden (1984, p. 312, emphasis added)

But although Boden represents herself here as being opposed to "neo-Darwinism", it should be clear that there is, in fact, very little difference between the position she describes and the general position envisaged within the scope of evolutionary epistemology. Specifically, Boden seems to be assuming that neo-Darwinism must rely exclusively on generators of variation which are *unbiased*; but as I have already explained, that is a mistaken view.[23] The structure of Darwinian explanation (or, more generally, of evolutionary epistemology) demands only that, to whatever extent a generator of variation exhibits significant favourable bias, this will require a further, recursive, invocation of the UVSR principle; and this can bottom out only with generators whose explanation or genesis is, we may say, unproblematic.

[22] I have been unable to identify the relevance of Boden's citation here: (Arbib 1969a) makes no reference, that I can see, to automatic programming; it seems possible that the intended reference was actually to (Lenat 1983).

[23] While I believe that Boden is mistaken in her interpretation of neo-Darwinism, I also consider that she can hardly be blamed for this. Evolutionary biologists have not always been very clear on the issue, though there have been some useful recent discussions (e.g. Dawkins 1989a; Wills 1991). I review this in more detail in (McMullin 1992b).

Boden evidently accepts all this: thus she says, variously that "the initial heuristics must evolve by random mutation (since there is no suggestion of teleology here)" (p. 312), and that "a structural theory can even allow that contingency is sometimes *essential* to creative intelligence" (p. 314).

Nonetheless, a literal reading of the full passage quoted above might still suggest that Boden has something stronger than this result in mind: the last sentence, in particular, with its reference to favourable mutations becoming "increasingly probable", could be read as implying some kind of *inevitable* progression, or even acceleration, in the growth of "adaptation" (or, in my terms, in the growth of knowledge). That would obviously be deeply contrary to the principles of evolutionary epistemology. But, I doubt that Boden herself really intends such a strong claim, for there is no explicit *argument* to such an effect. I shall not, therefore, consider her analysis any further.

By contrast, I think that Lenat (1983—a work which is heavily referenced in Boden's discussion) *does* have a genuine, if implicit, disagreement with the principles of evolutionary epistemology; but, equally, I believe that Lenat is wholly mistaken in this. Lenat's epistemology seems, at that time at least, to have been a naïve inductivism:

> The necessary machinery for learning from experience is not very complex: accumulate a corpus of empirical data and make simple inductive generalizations from it.
>
> Lenat (1983, p.287)

Lenat was evidently unaware that the idea of induction presented any difficulties *in principle*.

I should note that Lenat was working on a system (`EURISKO`) which went some way toward meeting the architectural requirements I have identified for realising AI. Specifically, `EURISKO` included components ("heuristics") for generating new conjectures, and the system was *reflexive* in the sense that these heuristics could operate on each other. However, the heuristics seem to have been relentlessly inductivist, and `EURISKO` cannot be viewed as implementing UVSR in any reasonable sense. In any case, the system had very limited success, and Lenat himself subsequently abandoned this line of research; but he has not, apparently, abandoned an essentially inductivist epistemology. In a recent discussion of the on-going `Cyc` project, he outlines the following objective for the system (to be achieved by late 1994):

> Demonstrate that Cyc can learn by discovery. This ... includes deciding what data to gather, noticing patterns and regularities in the data, and drawing from those patterns useful new analogies, dependencies, and generalizations.
>
> Lenat & Guha (1990, p. 357)

Thus, given that Lenat has still not recognised, much less analysed, the philosophical problems underlying induction, I suggest that any criticism of evolutionary epistemology implied in his work can be safely neglected.

To close this review of philosophical work bearing on the relation between UVSR and AI, I shall consider the position of John Holland and certain of his co-workers.

Holland is an engineer and scientist who has long been concerned with the problems of developing artificial "adaptive systems", and was in the vanguard of those advocating an evolutionary approach to such problems (e.g. Holland 1962b; 1962a). In particular, as noted earlier, Holland is the inventor of the so-called *Genetic Algorithm* (Holland 1975), a general purpose "learning" or adaptive procedure inspired in certain ways by the mechanisms of biological evolution. Holland has specifically attempted to apply the Genetic Algorithm in the development of machine learning systems which could overcome the brittleness of conventional expert systems (Holland 1986). It might seem, therefore, that Holland would surely support the position I have been advocating—that reflexive, heterarchical, UVSR processes are essential to the efficient, open ended, growth of knowledge. It transpires, however, that that would be, at best, an oversimplification.

In originally introducing the Genetic Algorithm, Holland identified himself as being concerned with the growth of "adaptation" (Holland 1975, Chapter 1). More recently, Holland has written explicitly in terms of the growth of "knowledge", particularly in the volume (Holland *et al.* 1986), co-written with K.J. Holyoak, R.E. Nisbett and P.R. Thagard; here they jointly identify themselves as concerned with "all inferential processes that expand knowledge in the face of uncertainty" (p. 1).

However, the situation is rather more complicated than this. The philosophical framework underlying Holland's approach is, at first sight at least, quite incompatible with that which I have been advocating: like Lenat, Holland *et al.* seem to be self-declared *inductivists*—indeed, the title of their book (Holland *et al.* 1986) is actually *Induction*. Despite this, I shall *not* be arguing against the epistemology, as such, of Holland *et al.*; instead, I shall suggest that the appearance of disagreement is mistaken, a matter of words and emphasis rather than substance. But, in the very process of reconciling these apparent differences, I shall conclude that the aspects of the growth of knowledge which Holland *et al.* choose to concentrate upon are largely those which I choose to neglect, and *vice versa*.

The *appearance* of disagreement between the position of Holland *et al.* and that of Popper and Campbell is clear enough. We find, for example, an unqualified rejection of "evolutionary epistemology" (Holland *et al.* 1986, p. 79; they explicitly cite Campbell 1974a; 1974b); and while we (eventually) find an admission, mentioning Hume and Popper, that the very possibility of induction is problematic, this is immediately passed over with the statement that "most [philosophers] have attempted to solve the narrower problem of determining under what circumstances the [inductive] inference can be justified" (Holland *et al.* 1986, p. 230).

However, I do not think the situation is quite as it may seem when selectively quoted in this way. While the writers apparently *believe* themselves to be opposed to Popperian epistemology, their genuine familiarity with Popper's work can reasonably be questioned. There is only one explicit citation of Popper (on p. 328, to the first English edition of *The Logic of Scientific Discovery*—see Popper 1980). This is followed, quite shortly, by an ascription to Popper of the view that any reluctance to abandon functioning (but refuted) scientific theories must represent "an irrational, egotistical attachment" (p. 332); Holland *et al.* go on to suggest (apparently as a contrast to this "Popperian" view) that, in practice, theories can only be discarded "when a new, alternative theory that does not require special assumptions arises and offers to replace the existing theory" (p. 332). But one could hardly find a more genuinely Popperian statement that the latter: it is, precisely, the notion of survival between competing hypotheses. Popper (1974b, p. 995) has had occasion to defend himself against a similar "criticism", where he gives a more detailed rebuttal. For my purposes it is sufficient to note that the rejection of a “Popperian” epistemology by Holland *et al.* may be more imagined (on their part) than real.

In fact, it seems that the “inductive” processes with which Holland *et al.* are concerned may be more or less identified with the “plausible-generate-and-test” processes of Boden (1984). Thus, in introducing their problem, Holland *et al.* quote C.S. Peirce at length to the effect that the growth of knowledge involves something like “special aptitudes for guessing right” (p. 4); and later (p. 79) they explicitly refer to this as “Peirce’s problem of generating *plausible* new rules”, which is almost identical to Boden’s formulation (though the latter is not actually cited). This last reference to Peirce is the more ironic since it is immediately juxtaposed with the already mentioned dismissal of (Campbell 1974a)—a paper in which (pp. 438–440) Campbell carefully and critically reviews Peirce’s profound ambivalence on the issues at hand.[24]

So: my conclusion here is essentially the same as previously outlined in reviewing Boden’s work. I hold, essentially, that the processes which Holland *et al.* describe as *inductive* are processes of unjustified variation. Notwithstanding my use of the term “unjustified” here, I quite accept that, in given circumstances, some such processes may do “better” than others (in the sense of generating conjectures which are “closer” to the truth). Similarly, I accept that the formulation and comparison of processes in this respect is a genuine and difficult problem. But, crucially, I hold that there can be no final or definitive “solution” to this problem; all “inductive” processes are heuristic and fallible; there is no “logic” of induction. I say this without doubting, for a moment, that even partial solutions to this “problem of induction” may be very interesting, and, indeed, pragmatically useful. It seems to me that Holland *et al.* do not ultimately disagree with any of this, and that their analysis need not, therefore, be considered any further here.

[24] I might also add that Popper himself has made very positive remarks regarding the philosophy of C.S. Peirce (e.g. Popper 1965, pp. 212–213); however, these remarks do not bear directly on the issues under discussion here.

3.9 Conclusion

In this chapter I have argued that computational systems *can* be said to have knowledge, in a perfectly conventional, biological, sense; that this knowledge can grow *only* via some kind of Darwinian, UVSR, processes; that such processes will therefore be an essential component of any system pretending to human-like intelligence (as represented, for example, by Turing Test performance); and that, in any case, such processes (as opposed to pure "Knowledge Engineering") may well be essential to the *initial* construction of any system which exhibits, or aspires to exhibit, human-like intelligence.

It follows that the realisation, in a computational system, of UVSR processes incorporating an open-ended, reflexive, heterarchical, architecture—which is to say, in effect, some form of *Artificial Darwinism*—is now seen as being at least *an* essential element (if not *the* essential element) of a serious AI research program. The next chapter will be devoted to reviewing some issues which arise in practical attempts to do this. I shall leave the final summarising word for the present chapter with Popper:

> I do not really believe that we shall succeed in creating life artificially; but after having reached the moon and landed a spaceship or two on Mars, I realize that this disbelief of mine means very little. But computers are totally different from brains, whose function is not primarily to compute but to guide and balance an organism and help it to stay alive. It is for this reason that the first step of nature toward an intelligent mind was the creation of life, and I think that should we artificially create an intelligent mind, we would have to follow the same path.
>
> Popper & Eccles (1977, Chapter P5, p. 208)