To appear in Proceedings Turing90 Colloquium
Edited by Peter Millican, Published by OUP

BEYOND TURING EQUIVALENCE(∗)

Aaron Sloman

School of Computer Science, & Cognitive Science Research Centre

The University of Birmingham

A.Sloman@cs.bham.ac.uk

Abstract

What is the relation between intelligence and computation? Although the difficulty of defining 'intelligence' is widely recognized, many are unaware that it is hard to give a satisfactory definition of 'computational' if computation is supposed to provide a non-circular explanation for intelligent abilities. The only well-defined notion of 'computation' is what can be generated by a Turing machine or a formally equivalent mechanism. This is not adequate for the key role in explaining the nature of mental processes, because it is too general, as many computations involve nothing mental, nor even processes: they are simply abstract structures. We need to combine the notion of 'computation' with that of 'machine'. This may still be too restrictive, if some non-computational mechanisms prove to be useful for intelligence. We need a theory-based taxonomy of *architectures* and *mechanisms* and corresponding process types. Computational machines my turn out to be a sub-class of the machines available for implementing intelligent agents. The more general analysis starts with the notion of a system with independently variable, causally interacting sub-states that have different causal roles, including both 'belief-like' and 'desire-like' sub-states, and many others. There are many significantly different such architectures. For certain architectures (including simple computers), some sub-states have a semantic interpretation for the system. The relevant concept of semantics is defined partly in terms of a kind of Tarski-like structural correspondence (not to be confused with isomorphism). This always leaves some semantic indeterminacy, which can be reduced by causal loops involving the environment. But the causal links are complex, can share causal pathways, and always leave mental states to some extent semantically indeterminate.

1. The Problem

Artificial Intelligence has revolutionised the study of mind by introducing computational theories of mental processes. Many people believe that computational concepts and theories suffice to explain the nature of perception, learning, creativity, motivation, emotions, feelings, consciousness etc. because all these are thought to be computational states, events or processes. Indeed, I once put forward such a view (Sloman 1978). But this presupposes that we know what computations are. Do we? Do analog computers do computations? The concept has become less clear with the growth of interest in 'neural computations'. I shall try to show that we have the following options:

a. Define 'computation' in the logico-mathematical sense (Turing-equivalence), but accept that in this sense computational mechanisms are not central to explaining intelligence.

b. Extend the definition to include all the kinds of processes that play a role in intelligence. It is then hard to draw a line between computational and non-computational processes, without falling into either circularity (computational processes are the ones needed for intelligence) or triviality (all processes are computational).

(∗) Revised April 1995

c. Abandon the idea that any precisely defined concept of computation can be the *key* notion underlying intelligence, and instead study different kinds of architectures and mechanisms to learn what kinds of roles different sorts of machines can play in the design or explanation of intelligent systems.

In support of (c) I'll explain why most attempts to define 'computation' either fail to define a concept adequate to explain intelligence, or else fall into circularity or triviality. I'll then expand further on option (c) and end with a discussion of how to give semantic capabilities to computational and non-computational machines. All this is not an attack on AI, but a recipe for progress.

2. The formal concept of computation

The mathematical concept of 'computation' is the only well defined concept of computation. It is concerned with purely formal structures. This point can be obscured by the process/product ambiguity. A process of computation may produce a *trace* for example a long division presented on paper. Both the process and its enduring trace can be called computations, but in different senses. The formal concept of computation involves no notion of process, causation or time, and is concerned only with the structural properties of such traces, no matter how they are produced. (Similar process/product ambiguities are associated with: 'proof', 'derivation', 'calculation', 'analysis', 'design', 'construction'.)

Formally a computation is a discrete sequence of structures satisfying certain conditions: i.e. every item in the sequence is either part of an initially given set of structures or derivable from earlier items according to fixed rules. The structures may be machine states, expressions in a logical language, or other configurations. Whether the sequence is physically embodied or merely an abstract structure is irrelevant, as is any question of what sort of mechanism, if any, produced it. In this sense leaves blown about on the forest floor could form a computation, e.g. solving an equation, as long as there is a description of the patterns formed by the leaves under which they conform to appropriate rules.

Not even physical embodiment is required. Using the unique factorisation of integers as powers of primes, Gödel showed how formulas of arbitrary complexity can be systematically encoded in integers. (See Nagel and Newman, 1958 for more details.) A complete (finite) computation, no matter how complex, corresponds to a sequence of Gödel numbers, which itself can be expressed as one large Gödel number. Thus a number can satisfy the formal conditions for being a computation. An infinite computation, e.g. producing the decimal expansion of π, would correspond to an infinite sequence of Gödel numbers.

This formal notion of computation, equally applicable to physical processes and non-physical mathematical structures, does not on its own enable us to build useful engines or explain human or animal behaviour. An abstract instance of computation (e.g. a huge Gödel number) cannot make anything happen. This shows that being a computation in the formal sense is not a *sufficient* condition for being an intelligent *behaving* system, even though the formal theory provides a useful conceptual framework for categorising some behaving systems. For instance, it establishes limits to what is possible and provides a framework for studying space/time complexity requirements and trade-offs. For the purposes of construction and explanation of intelligent systems, we need to combine computational ideas with the idea of a machine with causal powers. Smith (1988) argues, mistakenly I believe, that 'causation' is already part of the concept of computation. I have shown (Sloman, 1986, 1992b) that confusion on this point leads to systematic ambiguity in the 'Strong AI' thesis, and in criticisms of it by Searle (1980) and others.

3. Is the standard concept of computation broad enough?

The class of finitely specifiable sequences of structures turns out to be the class of sequences that can be generated by a universal Turing machine. There are at least three types of processes that might be involved in human brains that would not necessarily map on to a single Turing machine: (a) asynchronous parallel processes with independently varying speeds, (b) continuous, or at least non-discrete processes and (c) physical processes that are not known to fit (or not to fit) the computational model, e.g. chemical processes in the brain. I don't know whether (c) is subsumed by (a) and (b) or whether other classes of exceptions to Turing equivalence exist.

Any collection of *synchronised* parallel computers can be mapped onto a Turing machine by interleaving their execution. However, for *un*synchronised variable speed parallel machines, this mapping cannot always be specified. The system does not have well-defined global states and well-defined state transformations, as a Turing machine does. Similarly if a system goes through non-discrete changes it cannot be modelled on a Turing machine, whose states form a discrete sequence. A non-discrete set (like the rational numbers) may be merely *dense* rather than *continuous,* if between any two different states there is always another. Continuity requires more than this, e.g. the existence of limits of bounded monotonic sequences. Non-discrete processes, whether merely dense or continuous, cannot be modelled on a Turing machine, for it cannot have a succession of states such that between any two states there occurs a third. A Turing machine can specify all the rationals, but cannot generate them in their natural order (as Zeno's paradoxes show). Of course, quantization allows any bounded non-discrete process to be simulated to any required degree of accuracy on a Turing machine, with a speed cost that depends on the accuracy. Chaotic processes may be an exception.

Not all processes that are used to solve problems, make predictions, draw inferences, simulate something, *necessarily* involve discrete determinate processes. Both the non-discrete variation of analog computers (or slide rules) and the random properties of quantum effects might be able to play a useful role in some kinds of calculations and simulations. Should we rule out such non-Turing processes from playing an important role in intelligence?

An example relevant to some kinds of image processing and physical problem solving is continuous rotation of an image. We often find it useful to rotate or slide sheets of paper and transparencies continuously relative to each another, e.g. in planning or problem solving. Digitised approximations to such continuous change are now commonplace in computer graphics, and have been used internally in AI programs (Funt 1980). But they are only approximations. In this case it is not clear that fully continuous processes would add anything, though some non-discrete processes when embedded in physical mechanisms of an appropriate kind can go much faster than a simulation on a discrete computer, which is why analog computers are sometimes used for speed.

It is an open question whether animal intelligence depends (in part) on brain processes using non-discrete variation, such as states of neurons. Chemical soups (e.g. drugs and hormones) can alter the character of mental processes). Are these all to be called 'computational'? Why stop there? Why not include digestion, the growth of a tree, or even the processes in a thundercloud?

It may be that all physical processes are discrete at some level. Any set of measurements made over a finite time will be discrete (Fields 1990). But for many purposes, like predicting the motion of planets, it is useful to conceive of them as non-discrete, e.g. so that differential equations can be used in making predictions. Is there any reason to rule this out in mechanisms underlying natural or artificial intelligences? Defining computation in terms of discreteness or Turing equivalence (Pylyshyn 1986, Haugeland 1985) excludes these things. But is this just an arbitrary restriction of mechanisms to be used in AI (for engineering purposes, or for explanatory purposes)? We could broaden the definition of ‘computation’ so as to include, for instance, devices for solving problems using non-discrete processes, like slide rules and analog computers, or soap bubbles stretched on frames to work out minimum stress designs. That would increase the range of tools available within the framework of a computational approach.

It may turn out that systems that appear to be varying non-discretely are actually discrete at some lower, sub-atomic, level, but if so that is an empirical fact: there is no *conceptual* reason to rule out essentially continuous mechanisms from playing a useful role in human-like mental processes. However, if we allow ‘computation’ to include possibly non-discrete brain processes, slide rules and soap bubbles then it is hard to see how we can draw a boundary between computations and non-computations. Why should we bother?

Even if all processes in intelligent machines are exactly mathematically equivalent to Turing computations, it remails possible that some processes cannot be simulated on a Turing equivalent machine except too slowly for practical use. Even if all can, there may be some important differences. For example, three synchronised machines doing the same task in parallel are mathematically equivalent to one machine, yet the difference in reliability is significant to an engineer.

So, if real intelligent agents require some processes that cannot be produced by a single physically embodied Turing-equivalent computer *within required constraints* (of time, cost, bulk, energy consumption, reliability, etc.) then the mere theoretical possibility of close digital approximation would be irrelevant to explaining how *actual* systems work.

A possible counter-argument is that *all* processes are computational. In (Sloman 1978) I speculated, half seriously, that all physical processes might turn out to be computational. Fields (1990) attempts to prove that *measurable states* of any observable physical process can be interpreted as the execution of an algorithm that could be run on a universal Turing machine. His proof assumes both quantum physics and the existence of a well defined mapping from events to times. These assumptions need not hold in all logically possible universes. Morever, if it were true that *all* physical processes were computational, then it would be trivially true, and therefore uninteresting, that intelligent systems are computational, though it would remain an interesting task to try to demarcate the special types of computations required for intelligent systems.

However, if, as physicists tell us,

(a) no physical system has a totally determinate observable state

(b) transitions from one state to another are probabilistic rather than deterministic

then even physical objects in themselves, as opposed to our measurements of their behaviour, cannot be treated as Turing-equivalent computers, since (a) and (b) contradict requirements for Turing machines. Actual computers are built so as to minimise the large-scale effects of (a) and (b). Failure to do this completely leads to malfunctions, though mechanisms such as self-correcting memory devices reduce their impact.

We can sum up so far as follows: (i) The only precisely defined concept of ‘computation’ is the mathematical concept. This is a purely structural concept. (ii) This concept is in some ways too general and in other ways too narrow to provide a general framework for studying all possible mechanisms underlying intelligence. (iii) It is not clear, however, that we can find an alternative useful general definition that covers all the interesting cases and avoids the twin traps of circularity (because it presupposes some aspect of intelligence) and triviality (because it implies that all processes are computations).

Let us look at a few alternative attempts at defining ‘computation’: as a functional concept, as symbol manipulation, and as rule-governed processing.

3a. Computation as a functional concept

Clark (1988) proposes that instead of defining computers on the basis of their intrinsic properties we should do so on the basis of their origin and use: if a mechanism was designed or used for a particular sort of purpose or if it evolved biologically to serve certain needs of organisms then it might be computational, otherwise not. I.e. 'computation' is a functional concept like 'table'. In order to be a table something has to be used as one or intended for use as one: a rock in the wilderness with a table-like structure would not be a table unless used as such. Similarly any physical process has the potential to be a computation in the functional sense, e.g. if used (like a wind-tunnel) to solve a problem such as modelling some other object, to explain or predict its behaviour.

We would still, however, have to specify exactly what functions are *characteristically* served by computations. Digestive mechanisms that discriminate and analyse chemical substances evolved to serve animal needs. Are they computational? Visual perception involves intelligence, but is the eye-ball a computer, or part of one? Is it correct to say that the retina performs computations but not the lens? It is not clear that the functional analysis helps us answer such questions.

In the context of trying to explain intelligence as computational, it would be circular to define computers as those things that evolved, or were designed, for use by intelligent agents in processes like calculating, sorting, searching, perceiving, remembering, deciding. We need a characterisation of computational processes that enables us to use them to explain how intelligence is possible, not one that *presupposes* the existence of intelligent systems.

3b. Computation as symbol manipulation

Some people think of computation as *symbol* manipulation, where 'symbol' implies having a meaning. This also risks circularity; it is circular to assume that there are meaningful symbols to be manipulated, if we are trying to explain how meaning arises. In this context we must reject any definition of 'computation' in terms of manipulation of meaningful structures, and definitions in terms of making inferences or drawing conclusions.

3c. Computation as rule-governed processing

We could drop the reference to symbols, and simply define computations as 'rule-governed' processes. But this still risks circularity, if being *governed by* rules involves *understanding* the rules. Understanding is part of what we wish to explain: we must not assume it as a primitive. Can we avoid circularity by using a notion of being 'governed' that does not presuppose understanding, but might provide a basis for it? Computational processes would then be processes controlled by rules.

But if the rules are not understood, i.e. meanings play no role in the control, then the words 'rule' and 'governed' are misleading, and we are simply left with the notion of processes that are controlled by something. What, then, is control? Various forms of control — mechanical, hydraulic, electronic, chemical, etc., — have been studied under the heading 'control theory'. It seems acceptable to say that all computations are controlled processes, but are all controlled processes computational? Is the control of a spinning stone by its distribution of mass a computation?

Control is just a special case of the notion of causation: one thing controls another if there is some sort of causal relation between the controller and the controlled. (If it is a two-way relation, then the control includes feedback.) Analysing the concept of 'cause' is one of the hardest problems in philosophy (cf. Taylor 1992), but I shall, for the purposes of this paper, assume we all understand it.

We now seem to be moving towards a definition of 'computation' that is so general (e.g. computations are processes controlled or partially controlled by structures) that it includes everything, trivialising the claim that mental states are computational. We could try to avoid the triviality by limiting computation to cases where the kinds of controlling structures are more like computer programs. Unfortunately, as previously indicated, that would rule out too many candidates for inclusion as computations, for instance neural computations. Moreover, the notion of 'control by program' has difficulties, as I'll now show.

4. Do computer programs control processing?

There are several different kinds of control relation between a computer program and the behaviour it produces, some tighter than others. In particular, programs need not *totally* control the behaviour, since programs often run under operating systems that limit what they can do. On a time-shared computer each process is frequently interrupted by the operating system and suspended while another program runs. If a program attempts to access certain parts of memory or certain devices this causes it to be interrupted or aborted. Programs can have time bounds and space bounds. The 'permissions' and 'privileges' handled by modern operating systems impose various limitations of control by individual programs. So each program typically has only 'bounded control' over the processes it generates. (Cf. Sloman 1992b).

Many programs written by human programmers are not even partially in control of the processes they generate, at least not as a driver is in control of a car. Some programs are compiled to a different, machine-code, program, and after compilation the *original* program is not in control: no change in the original will affect the execution in the way that turning a steering wheel affects the motion of a car. Control by such programs is 'ballistic,' not 'online'.

A program has more direct 'online' control if it is not compiled but stored in the machine and 'interpreted'. But what is in control depends on a point of view. The program can be construed as actively controlling the process, or as a passive structure, used by the controlling interpreter. Even a compiled machine code program can be viewed as a passive structure used by the CPU to generate behaviour.

The 'ballistic' *vs* 'online' distinction is not the same as the familiar distinction between 'open-loop' and 'closed-loop' control. A machine code program's control is online, but whether it is open-loop or closed-loop depends on whether its behaviour depends on results of preceding behaviour, i.e. whether it uses feedback, in which case the control uses a closed loop. The loop may include the external environment (accessed through sensors and motors) or just internal states. A program with no conditional instructions and no parameters set at run time via sensors would be open-loop. A program that does not use any feedback is open-loop. It may nevertheless be online if it is in direct control of behaviour.

So the attempt to define a non-circular, yet sufficiently general, notion of computation in terms of control by structures requires clarification of the amount and kind of control required, as well as what sort of entity may be in control. We can't require a separable stored program using an explicit programming language, if we wish to include neural computations - unless we stretch the notions of program and language to encompass the topology of neural interconnections and connection weights. If we are prepared to say that a neural net constitutes a program, then what isn't a program? Why not include *every* physical or chemical structure that controls the behaviour of something?

5. Control by virtual machines

Another problem with compiled programs is that their instructions specify quite different actions from those referred to in the original high level programming language: there is a *semantic mismatch.* Programs written by human programmers usually refer to ‘virtual’ structures and operations on them, e.g. arrays, lists, records, trees, graphs, data-streams, or numbers, whereas the machine code instructions will generally refer only to operations on bit patterns and locations in a linear array of bit patterns.

Can instructions of the latter kind control processes of the former kind? Could we be mistaken in describing computers as manipulating arrays, lists, and other abstract data-structures? Do they really only manipulate bit patterns? Bit patterns themselves, however, are also virtual structures: what the *physical* machine contains is not bit patterns but wires, switches, voltages, currents, etc. Even these can be thought of as virtual mechanisms implemented in still lower level quantum physical processes.

The word ‘virtual’ as used here (as in Computer Science) does not contrast with ‘real’. (The word ‘abstract’ might be more appropriate.) Neither does it have anything to do with so-called ‘virtual reality’ systems, which are convincing simulations of physical processes. Virtual processes in virtual machines really exist and can have causal consequences. They are virtual in the sense that their contents are high level abstractions which are *implemented* in a ‘lower level’ machine. It could be argued that the kinds of physical mechanisms we normally talk about, e.g. lawnmowers, printers, are virtual machines in this sense, i.e. implemented upon lower level machines that only physicists know about. Whether there is a ‘bottom layer’ is an old metaphysical question, to which I offer no answer here.

Some readers may be inclined to argue that there is only one ‘ultimate’ level of reality and only at that level can processes be controlled. This requires drastic rejection of most common sense concepts of causation and control. Moreover Taylor (1992) produces arguments for liberal applicability of the concept of ‘cause’ to any type of domain. I shall continue to talk as if structures in a virtual machine can control processes in a virtual machine: i.e. virtual machine states and processes can have causal powers.

Most useful computations are not spatially located physical processes, but virtual processes in virtual machines, e.g. alphabetically sorting a list of words, which rearranges pointers but does not physically move anything. Such processes can have causal roles. E.g. re-ordering a list of words can change the order in which external actions are performed. Similarly a virtual machine event, like adding a word to a sentence in a word-processor can cause other virtual machine events, like rearranging page boundaries. These are causal relations between states and events in virtual, not physical, machines, though their implementation depends on physical causation. So do social, political and economic causation. In all cases, there are true counter-factual conditionals of the form: *if X had not occurred Y would not have occurred,* and *as long as X occurred, Y would have occurred even if Z had also occurred.* This justifies attributions of causality whether in physical machines, socio-economic systems, the human mind, or software systems.

Some people find it hard to accept that processes in virtual machines can have real causal powers, because they think there is a ‘bottom’ level physical reality which is the only realm in which causes can operate, for which I know no good argument. If true, that would imply that most of the things that we regard as causes are not real causes. That would be disastrous for normal human planning, the operation of the law courts, software engineering, and so on (Sloman 1994c).

So, when a compiled, machine-code program is controlling behaviour at the bit level, there is also usually another virtual program controlling behaviour of a higher level virtual machine at a level of abstraction closer to the original source code. (Such a program can change itself at run time so that it no longer corresponds to the original.) When a compiler performs optimising transformations it may not be clear exactly *which* virtual machine is implemented in the resulting code. But that does not mean there isn't one. Sometimes there are several layers of virtual machines: implementation is often multi-level. So we may have to allow our intelligent system to include some high level virtual machines.

6. We need a better set of concepts

We have many ill-defined concepts that evolved for purposes of everyday life, but do not suffice for more sophisticated purposes. When we try to apply these concepts in new situations to make fine, previously unnoticed, distinctions we discover that they are unsuited for the job. This often happens in the development of science or a culture, leading to significant conceptual change. A striking example is the way concepts of kinds of matter developed under the influence of atomic theory, through the periodic table and theories of chemical composition. We need a similar revolution in concepts of mind and control.

The concept of computation played a useful catalytic role in extending our understanding of some intelligent processes (e.g. proving theorems, planning, parsing sentences), and has opened our minds to much wider classes of (virtual) mechanisms than were previously dreamed of. Nevertheless, for reasons given above, we do not yet have a notion of computation adequate to provide a non-circular, explanatory foundation for AI and cognitive science: (a) because the formal concept needs to be combined with some sort of notion of causation, control or machine, and (b) because the forms of processing in intelligent systems may go beyond computations that fit the Turing-equivalent model.

There is no sensible place to make a sharp boundary between computational and non-computational mechanisms, except by using one of the standard formal definitions that are all mathematically equivalent to the definition of 'computation' as what a Turing machine can do. This definition is inadequate for the purpose of identifying the central feature of intelligence, because satisfying it is neither sufficient nor necessary for a mechanism to be intelligent.

(a) It is not sufficient because the mere fact that something is implemented on or is equivalent to a Turing machine does not make it intelligent, does not give it beliefs, desires, percepts, etc. Moreover, a computation in this sense can be a purely static, formal structure, which does nothing.

(b) Turing machine power is not necessary for such aspects of intelligence as the ability to perceive, act, learn, make plans, have desires, feel pain, or communicate, because is no evidence that animals that have these abilities also have Turing equivalent computational abilities: human beings, for example, appear to be limited in the depth of recursion that they can use in understanding sentences. Moreover, with or without external memory aids in their calculations, they often make mistakes that no Turing machine would make.

As argued above, there is no apriori reason to rule out the possibility that non-Turing equivalent processes play a useful in human brains. In any case, I argue (Sloman 1992b, 1993, 1994c,) that *architecture dominates mechanism.* The particular mechanisms used are not as important as the global organisation of sub-systems with different functions. This is what determines the main capabilities of the system. By comparison, varying the mechanisms, for instance, switching between neural and symbolic mechanisms, or introducing non-computational mechanisms will have a marginal impact on the main capabilities of the system..

Because the relevance of computation to intelligence is so problematic, I suggest that we should keep an open mind and examine the potential uses of all potentially relevant mechanisms, whether computational in any sense or not. Then the task of a theory of mechanisms for intelligence is to analyse similarities and differences between mechanisms of all kinds and their implications for both explaining and replicating animal and human capabilities. Computational mechanisms may turn out to be relevant only to an subset of human capabilities. (Sloman 1994a, 1995a describe a process of exploring mappings between 'design space' and 'niche space'.) All this is not an attack on AI, but an attempt to generalise it, and to defuse attacks on AI which claim that computation will not suffice for intelligence. We can respond by asking "Show us the mechanisms that do suffice, and we'll add them to the AI toolkit".

7. The variety of 'computation-like' processes

There is a wide variety of machines whose internal and external behaviour is controlled by internal structures. We can call a machine (or its behaviour) more or less 'computation-like' according to how close it is to implementing Turing equivalent powers. This is not meant to be a precise measure.

A special case is a conventional computer 'obeying' a machine-code program expressed as bit patterns. This provides bit-manipulations in a virtual machine controlled by a structure in the same virtual machine. The controlling structure is a set of bit patterns interpreted as instructions, addresses and 'data' bits. The computer has a primitive grasp of the semantics of its machine language because of combined effects of (i) structural properties (e.g. relationships between bit patterns and an ordered set of locations) and (ii) the underlying design which provides causal links between bit patterns and such things as: internal actions, locations in the machine, and counting operations (Sloman 1985b). The combination of structural and causal links provides a foundation for meaning. This is a recurring pattern.

The machine *understands,* in a limited fashion, instructions and addresses composed of bit patterns. While obeying instructions it manipulates other bit-patterns that it need not understand at all, although it can compare them, copy them, change their components, etc. This limited, primitive understanding provides a basis on which to implement more complex and indirect semantic capabilities, including much of AI (Sloman 1985b, 1987a).

AI need not restrict itself to processes running on conventional computers. Biologists talk of development of an embryo as a computation (with DNA as program); and it is now commonplace to regard networks of neuron-like mechanisms, as performing computations. We can also think of the biosphere and natural selection as performing computation-like processes that produce new genes and gene combinations. These all illustrate the admittedly vague concept 'computation-like' which subsumes normal computer processes but is not committed to Turing-equivalence, as long as processes are controlled by some structure.

If embryo development, neural processes biological evolution, and conventional computer processes are all computation-like, is there any process that is *not* computation-like? This request for a well-defined boundary should be resisted. In deep science good definitions arise from good theories, which we still lack. We should study the variety of cases, their similarities and their differences. Trying to understand many small but real boundaries and their implications is more fruitful than arbitrarily imposing binary divisions.

When you push the corner of a table, all the atoms communicate with their neighbours, adjusting their own motions and mutual relationships as needed to fit in with what the neighbours are doing. The processes are at least partly under the control of the structure of the table. Is that structure a sort of program? Is the applied force a sort of input to the program? Is the process computation-like? The best answer may be that it is a special case - just as a circle is a special case of an ellipse, lacking some properties of other ellipses. Once this is clear, any debate about whether a circle is or is not an ellipse loses interest: in the mathematical sense it is an ellipse, whereas in a colloquial sense it is not. We can then ask exactly how this special case differs from others. Similarly with special cases of computation-like processes.

What other computation-like processes are there? How are the more interesting varieties different from motion of a table? One important difference concerns structural variability, which includes changes in complexity as opposed to quantitative changes in a fixed set of dimensions. Another involves differences of functional roles within a larger architecture. I'll try to explain both.

8. Steps towards a classification of behaving systems

8a. Variability of states

One reason why computers are so useful is that their digital memory structure supports enormous variability of sub-states: there are astronomically large numbers of significantly different internal states between which the machine can change. In computers and brains this is achieved by having large numbers of independently switchable elements. N switches each with K possible states permit K to the power N possible total states. If N is large, even small values of K (e.g. 2) produce unimaginably large sets of possible states. The possible sequences of such states form even larger sets, supporting enormous diversity of behaviour. If the elements can vary non-discretely an even larger set of significantly different states can exist, with non-discrete transitions (though it is not clear whether this is important for intelligence). However, continuously varying systems may not be able to control themselves reliably over diverse trajectories in state space. Continuous variation makes it harder for control mechanisms to discriminate their own states reliably. Thus, even if continuous variation plays a role, it may be a restricted role.

8b. Redundant state transitions in serial systems

A difference between a brain and a simple computer is that in the latter a single processor has to change all the units. Whether it changes 16 or 32 or 64 bits at a time, certain state changes cannot be done in one step, but have to go through intermediate states, whereas a parallel network of processors can all change at once. Whether the need to traverse useless intermediate states is a serious limitation will depend on such things as the speed of the processor, ways in which intermediate states can cause errors, and so on. Contrary to current philosophical fashions I do not believe that the differences between a neural net and a computer are deeply significant from the point of view of designing an intelligent system, though they may affect engineering requirements like speed and robustness: useless and potentially undesirable intermediate states occur in one but not the other.

8c. The importance of 'structural' variability

Many physical systems, and their variation, are usefully represented in terms of a point moving in a high dimensional vector space. Richer kinds of variation, allowing changing complexity, are required for biological and intelligent systems. For example, biological growth is not merely a process of changing measurements: it can produce an entirely new structure, such as an eye or wing, or a new relationship such as connection via a nerve fibre. Many kinds of human learning, e.g. learning a new language or learning algebra, seem to require structural variability, unlike, for example, the fine-tuning of a pre-existing motor skill by practice.

Similarly, states of understanding different sentences cannot easily be represented by changing values in fixed dimensional vectors of measurements, because of the potentially unbounded complexity of sentences in natural languages. There are limits to our ability to handle unbounded depth of nesting in sentence structures, but we can cope with long rambling sentences with variable numbers of significant sub-structures. (Compare the children's song: 'This is the house that Jack built'.)

Although the retina does not change in complexity as different scenes are viewed, visual percepts do vary in structure. The differences between seeing a ballet, seeing an Indian temple, and seeing a printed poem are *differences in the numbers and kinds of objects and relationships perceived,* not just *differences in values of a fixed size vector.* Different scenes need descriptions with quite different structures, and there does not seem to be any upper bound to the complexity of what might be seen: a square can have sides made of smaller squares, whose sides are made of smaller squares, whose sides....

State changes in biological and cognitive processes therefore involve creation, deletion or rearrangement of substructures. This *structural* variation is different from either continuous or discrete variation along a one-dimensional scale, or simultaneous variation in N different dimensions in a uniform multi-dimensional space. Structural variation can involve increasing or decreasing complexity. So, for a mechanism with structural variability, the space of possible states is not homogeneous: there are different kinds of neighbourhoods at different locations.

The amount and kind of structural variability in a system limits the kinds of interactions it can have with the rest of the world. A system that can have only N distinct states cannot distinguish N+1 kinds of environmental situations. However, the environment can be counted as part of the state, like the trail blazed by a forest dweller, (Sloman 1978, chapter 6.)

8d. Structural variability in virtual machines

Computer science and AI have shown how we can use a machine whose states are fixed-dimensional binary vectors to implement many new 'virtual' machines with very different kinds of sub-states and causal interactions. (Crucial to this is the use of 'pointers' — bit patterns interpreted as referring to memory locations.) A virtual machine with great *structural variability* can be implemented in a fixed dimensional *structurally invariant* lower level machine. Similarly, a fixed-structure neural net can implement a virtual machine providing structural variability, though usually at a great cost in speed.

AI has shown how computational systems can be connected to an environment via sensory transducers which trigger complex structural changes in internal sub-states, e.g. produced by changes in the environment or changes in the viewpoint. The internal sub-states so produced can in turn control behaviour in complex ways. Slight environmental differences can produce elaborate differences in behaviour. (Compare hearing 'Your mother is well' and 'Your mother is ill'.)

Sensitivity to the environment is a feature of many life forms, including growing plants. Not all organisms support rapid internal state changes. Not all support great structural variability. Not all can build a representation of the *structure* of an object or situation, as opposed to a collection of *measures* like temperature or chemical gradients. These are some of the important 'discontinuities in design space' that will have to be understood if we are to understand the evolution of human mental capabilities.

Discrete variability, including creation of new sub-structures and new connections between old structures, seems to be required for some aspects of human intelligence, mainly because discreteness can support long term stability (e.g. stored plans, or grammatical knowledge). But that does not rule out other kinds of mechanisms, including some with non-discrete variability.

8e. Independently variable, causally interacting, functionally differentiated sub-states

A finite state machine, like the ‘engine’ of a Turing machine, has only one unanalysable global state at a time. Other systems, like a Turing machine together with its tape, or a thermostat, have a collection of coexisting sub-states that can change independently and interact causally with one another . A collection of coexisting interacting sub-states defines an architecture. A special type of architecture is one in which behaviour is controlled by information (Sloman 1993, 1994c). In such a system there will be mechanisms for taking in information, for interpreting information, for transforming information, for transmitting information and for storing information (over varying time periods). The type and variety of information, and the kinds of uses to which it can be put, will depend on the architecture and the environment.

We have seen that sub-states may have fixed or changing structure, they may be physical or virtual, and causation (control) may be direct or indirect, bounded or total, online or ballistic, open-loop or closed-loop, ongoing or momentary, mediated by an ‘interpreter’ or physically direct. All these differences determine the kinds of functional roles available within the architecture. In some cases even the architecture can change. While a program is running in a computer the number of causally significant sub-structures (e.g. parse trees, or items in a database, or incrementally created sub-programs) can change drastically. So the number and variety of functionally distinct sub-states can vary. The same seems to be true of brains, even if the underlying sub-mechanisms are very different. The virtual machine architecture of an individual’s mind can develop over time.

By contrast, chairs, tables, slide-rules, amplifiers, and steam-engines, have nothing like the variety of causal interactions between different sub-states to be found in brains or the simplest modern computers. They lack the functional differentiation, the variety of causally distinct sub-states, and the ability to remove old or add new sub-states. This renders most physical objects unsuitable as major components in an intelligent system (though they may be used for restricted purposes as part of a more general system). A related criticism can be made of the once fashionable analogy between a brain and a telephone exchange. It is not just a matter of *degree* of complexity: a storm cloud has hugely varying internal states with intricate and changing patterns of causal interaction, but lacks the functional differentiation between states needed for intelligence.

It is not worth arguing over which mechanisms ‘really’ are computation-like, nor which are ‘really’ intelligent. We should instead ask which mechanisms are useful, or even necessary, for different kinds of intelligence. We should not expect that human intelligence requires exactly the same kinds of mechanisms as the capabilities of ants or mice or squirrels, though there is likely to be some overlap. For instance, both need the environment sometimes to control behaviour fairly directly, and sometimes indirectly.

To summarise so far: important differences between mechanisms are concerned with the number and kinds of independently variable sub-states that can co-exist at one time, the kinds of variation of the sub-states, the kinds of interactions between sub-states, whether the number of sub-states is fixed or can change, and so on. These are all *architectural* issues, concerned with functional differentiation within a global design. Different sorts of sub-mechanisms might be used to implement the same architecture for a higher level virtual machine, just as different electronic components can implement the same amplifier circuit.

8f. Functional differentiation of mind-like sub-states

In an intelligent agent, different sub-states have different causal roles in the total system. For example, we can distinguish 'belief-like' and 'desire-like' states if the system can apply a 'correspondence' test comparing sub-states with the environment, and failed correspondences generate different compensatory actions for different classes of sub-states. If S is a belief-like sub-state, failure of correspondence tends to cause the S state to be changed until the test succeeds, whereas if S is desire-like sub-state, the same correspondence failure initiates processes that tend to change *the environment* until the test succeeds. Note that for some systems the environment will be partly internal. In more complex systems, there will be far more than two types of controlling sub-states (Sloman 1993). For example planning will require 'supposition-like' control states.

8g. Semantic correspondence without isomorphism

It should not be assumed that the 'correspondence' mentioned above is anything like isomorphism. Neither sentences nor most pictures are isomorphic with what they say, describe, or depict. All that is required is some systematic way of checking a sub-state S against a perceived (internal or external) environment such that two outcomes are possible. One of them can be given a special status and called 'truth'. States for which the check gives that special result are said to 'correspond' with the environment. In that sense numerals, '1', '2', '3' etc. can be checked against groups of objects by using a counting procedure. But there is no sense in which the numeral '10' is *isomorphic* with the groups that correspond to it. Requiring isomorphism between representing states and what they represent would require animals and robots to build internal replicas of the environment, with internal eyes to perceive the replicas, leading to an infinite regress.

9. Behaviour is not enough: objections to Turing tests

A question that causes deep philosophical disagreement is whether internal mechanisms are relevant to whether a behaving entity is intelligent or conscious or has certain mental states. One view is that since all we have to go on in judging other people to be intelligent is behaviour, nothing more should be required of machines than that they provide appropriate behaviour. This presupposes that our beliefs about other minds are based entirely in inference from evidence. I suspect this is a myth, and that evolution has imbued us with deep assumptions about the nature of agents and objects in the environment. What perceived behaviour does is merely determine details of *what* we believe about other minds, not *whether* we believe.

Evidence could never suffice, for any behaviour observed over a finite time could in principle be produced by indefinitely many very different mechanisms, including possibly a giant precomputed lookup table which would not have any intelligence: only its designer would. Even Turing-equivalent systems, i.e. systems that map the same inputs to the same outputs, may differ significantly in how they do it.

We thus have two views: (a) intelligence is merely a matter of what a system *does. (b) intelligence depends not only on what* is done, but also on *how* it is done. It is foolish to take sides in this debate, since the relevant concepts (e.g. 'intelligence', 'mind') are too ill defined, as Turing saw clearly (Turing 1950). Instead what we should do is analyse exactly the space of possible designs to understand the implications of various differences in architecture or underlying mechanisms.

10. Towards a classification of behaving systems

Having abandoned the precise, formal, idea of computation as the central key concept, and replaced it with the more general notions of architecture and mechanism, we can now start trying to classify behaving systems according to how they work, e.g. (a) how many different kinds of sub-states they can simultaneously support, (b) what kind of variability those states can have, (c) what their causal interactions are, (d) how much internal functional differentiation they have, and so on. The following questions can be asked about each design:

1. *How many independently variable (physical or virtual) co-existing sub-states can the machine support?*

   1.a. Is the machine able to extend the number or is it permanently fixed? (For some machines, answers to these questions will depend on the level of analysis in the implementation hierarchy.)

2. *What forms of variation do the sub-states admit?*

   2.a. Do they vary discretely, non-discretely, continuously, smoothly?

   2.b. Do different substates have different kinds of variability?

   2.c. Do the mechanisms support *structural* variation?

   2.d. Do the underlying mechanisms provide the kind of variability required for logical or propositional representations (fixed function symbols with changing arguments and vice versa, with hierarchical composition)?

   2.e. Can one sub-state be 'stored' in another and then later 'retrieved' on the basis of a partial match? (I.e. is there a flexible content-addressable memory?)

   2.f. Do the mechanisms allow new associations to be created? (An association between X and Y requires a mechanism that when presented with X produces Y.)

   2.g. Do they allow arbitrary tangled networks of associations, or only linear chains, or trees? E.g. does the mechanism allow the items linked by associations themselves to be associations, like a tree of trees? Does it allow the links to have structure, e.g. with complex labels?

   2.h. Do the mechanisms permit *implicit* sub-states, like the theorems implicitly stored in a logical database, or information stored in a sparse array? (Some of the virtual structures may have more components than the physical structures that implement them, making it futile to seek correlations between information states and physiology.)

3. *What kinds of causal interactions and functional differentiation does the machine support?*

   3.a. Which changes in states can cause other changes in states? I.e. which substates are causally linked?

   3.b. Does it allow internal variation of one sub-state to be finely controlled 'online' by another complex sub-state (like a stored program controlling execution)?

   3.c. Does it allow the addition of a new sub-state S to trigger further sub-states, and so on? (Compare production systems, forward-chaining predicate logic systems, etc.)

   3.d. Are most of the changes controlled or initiated by the environment or is the majority of processing 'internally generated? In humans, mental states are continually changing, with or without external influence.)

   3.e. What kinds of internal feedback loops does it support? (Structural variability makes possible more than just positive and negative feedback.)

   3.f. Does it allow internal structures to be built that store information about other internal structures, or describe relations between them, for use by the machine? I.e. are there belief-like states concerning the machine's own internal state?

   3.g. Are all changes synchronised or can sub-states change at different independently varying speeds, i.e. asynchronously?

3.h. Can some states be belief-like, i.e. largely under the control of the environment, and others desire-like, i.e. able to generate actions when the environment does not ‘fit’ them. (See definition in section 9(c).)

3.i. Does the mechanism allow internal records of internal processes to be constructed for future use (in recollections, thoughts, etc.)?

(This is just a small sample of a wide variety of causal abilities and functional relationships to be investigated.)

4. *How fast can states change and causes propagate, relative to the kinds of changes that can occur in the environment?*

   Speed limits can have profound design implications. E.g. I’ve argued elsewhere (1987b 1992a) that certain emotional states arise out of mechanisms connected with resource limits.

5. *What is the global (high level) architecture of the system?*

   5.a Are there separable, but interacting, sub-systems each performing some complex function within the larger whole? What sorts of functions?

   5.b. Are the architecture and the collection of functions fixed, or can the architecture change through pre-programmed development or learning (as seems to be the case with human children when they learn new forms of self-control, or new forms of learning, or new reflective capabilities, or new sets of concepts)?

   5.c. Does the architecture support *semantic* capabilities, e.g. does it allow the machine to use some sub-structures to refer to others? (Discussed below.)

   5.d. Does the architecture allow different sorts of mechanisms to be used for different purposes, e.g. neural nets, production systems, etc.? (Hybrid systems may allow sub-mechanisms to be optimised for different tasks.)

6. *Are the different sub-states spatially separable,* i.e. embedded in different sub-structures (like computer data-structures) or superimposed in a distributed form (like superimposed wave forms or distributed neural states)?

7. *Is there always ‘cross-talk’ between the different sub-states,* so that changing one always has some effect on others, or can they be causally isolated?

8. *What kinds of internal and external self-monitoring can the system support, how reliable is it, and what can it be used for?*

9. *How much of the internal state is displayed externally, voluntarily or involuntarily?*

These are only some of the kinds of questions that can be asked about different sorts of architectures and mechanisms potentially relevant to designing or explaining intelligent systems. The answers will identify different classes of systems.

Some of the questions need to be made more precise, in particular the analysis of the causal roles characteristic of belief-like and desire-like states. That requires an exercise in the global design of intelligent systems. (Cf. Sloman 1987b, 1992a, 1993). The rest of this paper concentrates on the question how a system with independently variable sub-states can interpret some of its own states as having a meaning. No completely definite answer is to be expected because the question is inherently vague and ambiguous owing to the indeterminacy of our ordinary concepts.

11. The roots of semantics

If a mechanism can store and retrieve sub-states then people can use it as an information store, like a filing cabinet. But a filing cabinet does not understand anything. How can a machine treat some of its own internal substructures as referring to anything, i.e. as having a semantics for *the* *machine,* and not just for us? (This is the question that Haugeland 1985 posed in terms of a distinction between 'derivative meaning' and 'original meaning', which was later labelled by Harnad as the 'symbol grounding' problem.) Does a machine with 'original meaning', i.e. a machine for which certain symbols have a meaning independently of whether they have a meaning for anyone else, have to be a computational machine, i.e. capable of being completely replicated on a Turing machine? Could a purely computational machine support 'original meaning'?

In (Sloman 1985b, 1987a) I argued that there is no sharp distinction between systems that do and systems that don't understand the structures they manipulate. Rather there is a whole *cluster* of prototypical semantic capabilities in human beings, and different subsets of capabilities may be instantiated in different mechanisms - natural and artificial. Simple concepts from ordinary language, like 'understand' need to be replaced by a richer family of far more precisely defined concepts, related to underlying mechanisms, just as our concepts of kind of stuff evolved with advances in physics. 'Intelligent' is another such concept full of muddles: which is why I make no attempt to define it here. 'Consciousness' is an even worse swamp of confusions camouflaged by misplaced confidence that we know what we are talking about.

Some of the requirements for human-like semantic capabilities are structural (i.e. mechanisms are needed with certain capabilities) some functional (i.e. the mechanisms need to be used with certain roles within the whole system). AI work on understanding language or images hitherto has been more concerned with the structural than with the functional conditions: little or no attention has been paid to issues concerned with motivation, for instance.

I have shown (1985b) how some semantic capabilities can be found even in the way a digital computer uses its machine language. It uses some sub-structures (i.e. bit-patterns) to refer to locations in its (virtual) memory, some to possible internal actions (in a virtual machine), some to numbers, when counting, and so on. Of course the uses of semantic relations made by a simple computer are far more limited than the uses we make: for instance there is not yet motivation in the computer. Nevertheless, even now, it is the computer, not a person, that (a) uses the bit-pattern in an address register to determine which location is to be interrogated or changed, and (b) uses the bit pattern in another register to determine which action to perform. Filing cabinets were never like this.

Giving a machine a larger collection of semantic capabilities, with semantic competence closer to human abilities, requires both (i) a richer formalism than most machine languages (i.e. sub-states with richer kinds of variability) and (ii) a richer architecture, including both belief-like and desire-like roles (explained below). Such a virtual machine could be implemented in a much simpler physical machine.

The full story of the kind of architecture that not only begins to allow symbols to be related to the world by the machine, but also allows the meanings they express really to *matter* to the machine (as the news 'Your mother is ill' would matter to you), is very complex. It requires a theory of how different motivational processes work, on which more below. (See also Sloman 1987b and 1993).

12. Combining Tarskian semantics with causal links

There does not have to be any direct and simple mapping between representing structures and what they represent, as should be obvious from the fact that there are sentences containing disjunctions, negations and quantifiers, and pictures that have both a smaller dimensionality than the scenes they depict and often also a different topology: a typical 2-D picture of a 3-D wire-frame cube has more junctions than the cube has vertices, whereas a picture of an opaque cube has fewer.

Tarski (1956) showed precisely how a set M of real or possible objects can form a model for a set S of logical axioms, without M and S being isomorphic: e.g. a small finite set S, like Peano's axioms for arithmetic, may have an infinite model M. However, if M' is isomorphic with M, them M' is also a model for S, whatever M' may be. Thus Tarskian semantics can never determine a unique referent. (In general, not all models of a given set of axioms will be isomorphic with one another.)

If a mechanism has sub-states whose structural variability matches requirements for a logical formalism, with transformations corresponding to valid rules of inference, then Tarskian semantics will allocate an indefinitely large set of possible models to the sub-structures in the machine. These models may be either abstract mathematical structures or objects and relations in the world (e.g. a social system may model some set-theoretic axioms).

The set of possible models for S can be reduced by adding constraints in the form of new independent axioms, but this never suffices to pin down the model to a particular bit of the physical world; for there is always the possibility that some other exactly similar world, or portion of the world, provides as good a model as the intended one. Pure syntactic structure, however intricate, can never guarantee semantic definiteness (though uniqueness at a certain level of description may occur.

Semantic ambiguity can be further partly constrained if some of the sub-structures are causally linked with bits of the world. For example, electronic mechanisms ensure that bit-patterns in a computer are causally related to locations in its own memory rather than locations in another machine, despite having the same structural relations to both. Less direct causal links via the Internet can connect a more complex bit pattern (or symbolic address in a virtual machine) with a computer, or even a user, in a remote part of the world. In software engineering, unique semantic links of many kinds are set up by means of a judicious blend of structural mapping and causal linkage: without this mixture, electronic information services could not work. The use of structural mapping allows the causal links to be very loose (e.g. Internet connectivity is constantly changing). The existence of causal links removes, or reduces, the ambiguity inherent in the purely structural semantics.

Similarly, in AI systems, or animal brains, perceptual mechanisms and motors controlled by the internal sub-structures can set up causal links in both directions between internal sub-structures and aspects of the external environment. These links reduce the possible Tarskian interpretations of internal 'axioms', 'predicates', 'individual constants', etc. But they never totally eliminate semantic indeterminacy: in a rich and complex world we can always be surprised by unexpected ambiguities in our words and phrases.

Much work in epistemology and the philosophy of science has attempted to satisfy philosophical sceptics by eliminating such semantic ambiguity, but I see no reason to require guaranteed uniqueness of reference, as long as the mechanisms *usually* function adequately for the organism or agent concerned. In any case, I don't believe such guarantees can be achieved. Some philosophers will be unhappy about this, but neither human designers nor evolution, need adopt unsatisfiable requirements!

Can we extend Tarski's ideas beyond the case where the representing formalism is a logical (Fregean) notation (Sloman 1971, 1985a. 1995c)? We also need to allow different kinds of meaning that don't easily fit into Tarski's framework, e.g. emotive meaning.

A more general theory will allow a wider variety of structures, including non-discretely variable sub-states, to have semantics, while extending the kinds of semantic indeterminacy that can occur. Work on computer vision shows how the structural notion of semantics can be extended beyond logical representations. Here disambiguation by causal linkage is fairly direct, though semantic relations involving intermediate and high level interpretations (e.g. ‘X looks happy’) are complex and subtle. (Ballard & Brown 1982, Sloman 1989)

Human and animal visual systems and robots using TV cameras all seem to provide existence proofs that analogical representations (e.g. pictures or diagrams), and perhaps some non-discretely variable sub-structures, can have a useful semantics. Can we explain how? We can extend the notion of a portion of the world being a model for some representing structure in a machine by relating machine sub-states to the *roles* that they can play in a behaving system, and the ways in which these roles interact with the environment, via causal loops. Examples would be the different functional roles of bit-patterns and analog sensors in a robot. However, at present I can offer only an incomplete sketch of ‘loop-closing’ semantics.

13. Towards ‘loop-closing’ semantics

Let’s start with a thermostat and gradually add design complications. A thermostat connected to a room heater has a (primitive) belief-like sub-state that represents current temperature of the room, for example, the curvature of a bi-metallic strip. The thermostat also has a (primitive) desire-like sub-state corresponding to the required temperature setting. For each desire-like state D there is a range R of belief-like states such that if D and R co-exist the thermostat will make no attempt to change the environment. Otherwise it will turn the heater on or off. For each belief-like state B there is a set of possible states of the environment that will tend to produce B (when the sensors are working normally) and will tend to be produced by corresponding desire-like state D(B) when the heater is working normally).

For every sensor/controller pair there is an aspect of the environment whose variability matches the variability of the relevant internal sub-states. The match may be approximate. E.g. temperature settings may be discrete while the range of possible environmental temperatures is non-discrete (the reverse might be true in some machines, e.g. an analog representation of the number of objects in a container). Also the correspondence may not be one-to-one because of noise, lack of resolution, time-delays, projection from 3-D to 2-D, or other aspects of the measuring device or controller.

A thermostat whose behaviour depended on who was in the house, where they were, and how they felt, would require a far more complex set of internal states with more varied causal links. If the sensor produced not just a measurement but a structural description in a logical language then the causal links between the external and internal states could be very complex and indirect (e.g. going via a parser), and the sensing process could discard some incoming information (e.g. fine detail) and add other information (e.g. inferences about unobserved surfaces or likely behaviour of perceived objects).

13a. Loosening the links with belief-like states

Now consider a more complex controller that can separately measure and control a range of properties P1,P2,...Pn, of a machine or plant, but can only work on one of them at a time. For each such property Pi it has

(a) some kind of sensor Si that will produce or change the corresponding belief-like sub-state Bi,

(b) a settable control knob (or set of keys, etc) Ci that modifies the corresponding desire-like sub-state Di and

(c) an output-channel Oi that controls the relevant property Pi.

Suppose also that the machine can have only one such output channel turned on at a time, and has a selector that can switch between the different environmental properties to determine which is controlled.

For such a machine, sub-states still correspond to possible environments, except that it is no longer true *all the time* that each desire-like sub-state tends to change the environment to correspond to it. However, at any time when the i-th controller is disabled we can talk about the effect the i-th desire-like state Di *would have* in that context (including the belief-like state Bi) *if* the output channel Oi were selected. Similarly, if there are more possible desire-like states than output channels, then only a subset of the Di can be having an effect on the environment at any one time. This is clearly true of human beings: our legs, hands, mouths, etc. can be used to achieve different purposes at different times, but not all of them at once.

Likewise, if the different *sensors* can be temporarily disconnected, this suppresses the environment's tendency to influence belief-like sub-states, yet we can ask what the causal correspondence would be IF the particular sensor or controller were connected, and working normally. (The causal link is dispositional.)

Let us further loosen the connection between sensors and belief-like substates. There might be N different sensors, and K different belief-like states all derived (possibly via a neural net) from different combinations of the sensor readings, where K varies over time. So some of the sub-states are 'computed' on the basis of the signals received from several sensors, perhaps also using background information. An example could be a visual system building a 3-D scene description from a 2-D array of retinal information. If the process uses prior knowledge of the world, then that weakens the causal link between environment and belief-like states, e.g. when prior information is used to resolve ambiguities and reject some evidence (e.g. not seeing mistakes when proof-reading).

Some of the belief-like sub-states thus produced may be stored for future use, instead of being 'overwritten' as new information comes via the sensors. Then some effects of the environment would be long delayed. Similarly if desires lead to plans for future execution.

Causal links between the environment and the current set of belief-like states can be far more complex and indirect than in a thermostat, where the environment has direct online control of the belief-like state. The more complex and indirect the process that creates internal structures from sensory input, the more scope there is for internal malfunction and context-sensitive effects. Then the set of counterfactual conditionals linking the internal states to the environment becomes even more complex, and the correspondence depends less and less on direct causal links and more and more on structural properties of the internal states that constrain possible interpretations. This helps to explain the possibility of false beliefs, which cannot normally occur in a thermostat.

13b. Loosening links with desire-like states

Now, instead of a fixed set of desire-like states Di permanently connected to corresponding controllers or even a changing subset of Di directly connected to output channels, consider a high level virtual machine containing a *variable sized* store of desire-like states, created by 'motive-generator' rules, with context-sensitive 'motive-comparator' rules and decision-making rules for determining relative priorities of desire-like states, selecting a subset for action, retrieving or creating plans, and executing plans, possibly over an extended period with different plans interleaved if necessary. As with belief-like states, this extra complexity of processing (sketched in Sloman 1987b, 1992a, 1993, Beaudoin 1994) reduces the directness of the causal links between desire-like states and states in the environment. Instead of a simple discrepancy measure sufficing to turn control signals on and off (as in the thermostat) it may require quite complex internal processing of the relationship between belief-like and desire-like states, checking whether a desire-like state is 'satisfied', or not. Moreover, where a desire is not satisfied, complex planning and reasoning, making use of belief-like states, may be needed to produce appropriate control signals. Causation is often round-about.

There is clearly a huge variety of possible designs for mechanisms, some whose internal belief-like and desire-like sub-states are directly linked to the environment, and some where the links are very indirect, with varying numbers of intermediate stages in input channels or output channels and changing allocation of input channels and output channels to particular belief-like and desire-like states. Perhaps this provides one way of categorising the control systems of biological organisms.

13c. Loose causal links, and semantics

These complex designs undermine the notion that causal connections account for semantic relations. The loose and indirect causal links do not support finite detail of semantic relations. In thermostats where the belief-like states have dedicated input channels and the desire-like states have dedicated output channels, the semantic properties of different belief-like and desire-like states are determined almost entirely by their causal links. As we move away from such simplistic designs we encounter systems with more complex, loose and indirect causal links. Increasingly, semantic significance of their states will depend on *structural* as opposed to causal properties. I.e. we get closer to the Tarskian kind of semantics. However, we also find more and more scope for indeterminacy in the semantics, because of the weak and indirect causal links, and inability of structure to determine reference uniquely.

The more indirect and abstract semantics, together with generative capabilities in the mechanisms, can also explain the use of sub-states that refer to things remote in space and time or even to possibilities that are never realised (e.g. a bit-pattern addressing a non-existent location). This is an essential requirement for intelligent planning. The ability to give an internal sub-state a ‘supposition-like’ instead of a ‘belief-like’ role depends on the causal links with the environment being far less direct than in the thermostat.

I conjecture that biological evolution includes developments along the directions indicated here, with decreasing causal coupling of internal and external states going hand in hand with increasing structural complexity and functional differentiation of internal virtual sub-states, and longer term storage replacing online control. If these ‘design-space’ ideas can be used to distinguish different possible kinds of machines, perhaps they are also important for understanding different kinds of nervous systems?

14. Loop-closing models

Can we combine structural and causal ideas and specify a general semantic relation between sub-states in a behaving system and what they refer to. Consider an environment E containing an agent A, whose functional architecture supports belief-like and desire-like sub-states. Suppose A uses similar sub-structures for both, just as a machine can use bit-patterns both for addresses and for instructions. Then we can define the class of possible ‘loop-closing’ models for a set of structures S by considering a set of possible environments E satisfying certain conditions, when the action-producing mechanisms, the sensors, and the correspondence tests (section 8f, above) are working normally:

(a) States in E will *tend* to select certain instances of S for A’s belief-like sub-states.

(b) If Si is part of a desire-like state of A and E is in state Ei, A’s correspondence tests show a discrepancy between Ei and Si, then (unless A’s other belief-like and desire-like states interfere) A will *tend* to produce some environmental state Ej in E which *tends* to pass A’s ‘correspondence’ test for Si.

(c) If that happens Ej will tend to produce a new belief-like state in A.

So there are dispositional causal loops through which A's desire-like states tend to influence the environment, and the environment tends to influence A's belief-like states.

I repeatedly say 'tend to' to indicate that there are many additional factors that can interfere with the tendency, such as conflicts of desires, perceptual defects, accidents, wishful thinking, bad planning, and other common human failings. So these are very loose regularities, and cannot be taken to define internal states in any precise way. None of this presupposes that A is rational. It merely constitutes a partial specification of what 'belief-like' and 'desire-like' mean. However, a full specification will be relative to an architecture, within which functional roles can be defined more precisely.

More complex causal loops involving the environment will be involved in the way A's desire-like states are changed. This can involve internal motive generators, urgency, importance, and relative priorities, (Beaudoin 1994, Beaudoin & Sloman 1993). In simple designs, output is directly and continuously controlled by a discrepancy between desire-like and belief-like state, whereas in more complex cases the desire-like and belief-like states, together produce *chains* of actions, often specified in advance as part of the process of selection. In other words, advance planning is sometimes used. But not all architectures can support this. In those that do, the causal loops between states are more indirect.

14a. Causal loops and limited rationality

If A were completely rational and always had consistent motives and beliefs, then the tendencies mentioned above would be strict, whereas in real agents conflicts and errors occur, and more or less irrational behaviour is possible. In extreme cases people have to be forcibly restrained from harming themselves. Even when interactions between sensory input, belief-like and desire-like states, and motor output are not rationally comprehensible, this does not mean that internal states have no semantics.

For these reasons I think that any attempt to *define* mental states or process in terms of rationality or even approximate rationality is unacceptable. This undermines Dennett's notion (1978, 1987) of the 'intentional stance', which requires agents to be rational in order to have states with semantic content. The mechanisms and internal states that underly rationality can sometimes interact in bizarre ways to produce totally different results. We need a theory that explains both the rational and the irrational behaviour on equal terms, for instance an account of an architecture composed of interacting information processing subsystems, that sometimes function as required for rationality, but not always. Software engineers know how to build such systems, using not the intentional stance but the design stance applied to information-processing level descriptions (Sloman 1994).

14b. Local vs global semantic consistency

For a really complex agent with a large set of belief-like sub-states S there may be no possible environment providing a model for the *total* set S, because the current beliefs are globally inconsistent. Similarly the desires may conflict, with one another and with beliefs. People may want things which they know to be incompatible. The agent may be unable to detect all inconsistencies: for doing so reliably in large systems is computationally intractable, and therefore neither evolution nor robot designers can impose that requirement. In such systems semantic relations have to be local, or piecemeal: only fragments of the system have models, not the whole system. Perhaps this works because different subsets of the system are fully 'active' at different times, like the scientist who prays on Sundays or the kind father who bullies his employees.

14c. The irrelevance of history

The helplessness of human neonates tempts many to assume their minds are empty. This may be part of the strong tendency to require concepts to have been derived from individual experience. But a new-born foal can run with the herd within hours, and could not possibly have learnt all the required concepts of 3-D structure and motion and action. So if we accept that it sees and that seeing is an intentional state, the interpretive concepts underlying that state need not come from the individual's interaction with the environment: the interactions of long dead ancestors may suffice, as they do for the foal.

But even that cannot be a *logical requirement.* Suppose that by a highly improbable fluke of mutation an animal were born with the visual and action capabilities of a foal *without* this being the result of previous selective pressures. Would the new sport see or have intentions relating to the environment as well as a new foal? Surely the *current* internal structures, mechanisms and causal links would suffice, for all practical purposes, without the normal causal history. (Whether such a mutation is likely to happen is irrelevant.) (Compare Young 1994.)

No doubt some philosophers will retort: 'this animal does not ''Really'' see, or think, or take decisions, despite appearances, because it does not have the right evolutionary history'. We should resist disputes about essentially trivial matters of 'correct' definition. To avoid such disputes, we can define two notions of 'see': one of which (to seeH) requires a normal historical source, while the other (to seeA) is a-historical. Apart from that there is no difference in the details of the capabilities, i.e. how well they enable the organism to survive. Those of us who use 'see' to mean 'seeA', have a ready made way of talking usefully about new specimens whose perceptual ability is not rooted in evolutionary history. Those who insist on using 'see' to mean only 'seeH' will find it very awkward to describe.

14d. Semantics and inaccessible referents

A requirement that semantic relations depend on causal links that preserve a correspondence between representing and represented things obviously fails for semantic states referring to remote parts of the universe, the distant past, the distant future, unrealised possibilities, etc. These referents are not capable of directly engaging causally with current beliefs or desires, though they may be linked through counterfactual conditionals about what would happen if the agent had a different location in space and time. However, if one tries to work out what would be the case if the agent were sufficiently close to the remote place or time for direct causal interaction, it may be impossible to decide what else would be the case: an extreme example of the 'Frame' problem. For agents that are capable of using a generative notation with inference techniques, it may be better to define the possible models in Tarskian terms, using a generative compositional semantics, with the restriction that the models contain sub-models in the environment that act as causal loop-closing models, to select the right referents.

Not all organisms and machines have the internal architecture required for coping with this kind of semantic relation. For those that don't, any kind of semantics that they support will be simplified compared with human capabilities. Unlike Fodor (1987), we do not require all representing notations to be generative, although a system with generative capabilities will, of course, have more scope for creative intelligence and coping with unexpected situations, as well as thoughts and desires concerning remote places.

There need be no sharp boundary to the class of possible environments that are models of a sophisticated agent's beliefs or desires - i.e. the semantics for the internal states will be indeterminate in various ways. This in itself should not disturb us if we are interested in explaining human intelligence, since there is plenty of evidence that human languages (and probably internal representations too) are indeterminate in various ways. (E.g. how big is a big man? How much water must fall on a rainy day? Is a circle an ellipse? Is it 3 o'clock on the moon? Are liquids mixtures or compounds? Where are the boundaries between species of birds?) Semantic indeterminacy is part of the human condition. It may be unavoidable in robots.

15. Work to be done

The ideas presented so far concerning semantics are both tentative and lacking in precision. Considerable research is needed to clarify and extend them. In particular:

- Unlike the languages discussed by Tarski and most logicians there is no need for a fixed precisely delimited syntax to be used: intelligent systems can creatively extend the variety of representing structures they use, and humans frequently do this.
- The semantics assigned to particular notations by an individual need not be fixed: even one-off interpretations are possible (including 'lets pretend' games by children). It is tricky to fit this into an analysis that is very dependent on counterfactual conditionals, which, in turn, depend on lawlike generalisations. (This example of 'one-off' semantics refutes many philosophical theories of meaning!)
- The notation can include context sensitive elements whose semantic role needs special treatment, like the indexicals whose denotation depends on the instance of use: 'this', 'now', 'I', 'we', 'he', etc.
- Semantic relations may depend not only on an individual's mechanisms and functional architecture, but also a social system or culture involving other intelligent individuals. This can determine the 'scope' of concepts, like 'valley', 'healthy', 'marriage', 'war' and 'honourable'.

16. Conclusion

Although computation has had a powerful catalytic effect in extending our ideas concerning possible mechanisms, we can now abandon the notion that the concept of computation is the only or even the central foundation for the study of mind. Instead we need to look at a whole variety of architectures and mechanisms, from the design standpoint, to see what kinds of more or less mind-like systems they can support. This was option (c) defined in the introductory section.

I've offered the beginnings of a conceptual map, albeit still a blurred and incomplete one, into which we can fit many kinds of natural and artificial mechanisms and processes that could be useful for intelligent systems. Some will be closely related to our precise notion of computation, and some will not. Whether they are or are not is of little importance for the question whether they provide the kind of functionality required for various sorts of intelligent capabilities, except where we are studying intelligent systems with particular computational requirements, e.g. proving theorems, making plans, etc.

I have not argued (like Searle) that a digital computer cannot understand symbols it uses. I simply draw attention to the need to consider a broader range of machine types than simply Turing-equivalent machines, for the purpose of explaining or designing intelligent systems that can function as effectively as we do in our world. We also need a range of mental concepts corresponding to each of our everyday concepts like 'understand', 'refer', 'believe', 'desire', 'perceive', etc. The different concepts will be grounded in capabilities supported by different architectures.

This is not a philosophical argument about 'correct' concepts to use, but an engineering argument about the appropriateness of different (animal or artificial) designs for different tasks. Exploring such relations (mappings between design-space and niche space) is part of the goal of AI (Sloman 1995a). I claim that also provides the best framework for philosophy of mind.

I have shifted the emphasis away from computation towards a general notion of mechanism because it is hard precisely to define a concept of computation that is adequate as a non-circular, non-trivial, foundation for explaining mentality. So the hoped-for single boundary between computations and non-computations is replaced by sets of features of computation-like mechanisms defining a *variety* of design boundaries with different implications, which we still need to explore. The notion of computation can then be replaced by a new taxonomy of designs, covering a more general class of architectures and mechanisms, which may lead us both to consider new designs and to improve our understanding of old ones.

On this basis we can explore a variety of more or less rich semantic notions, some relatively closely tied to causal links with the environment, some closer to the structural relations defined by Tarski (not to be confused with isomorphism!) and some linked to functional roles within the architecture. Mixtures of these different kinds of semantics can be instantiated in machines with different sorts of architectures and mechanisms.

Showing in detail how different subsets of machine types can support different forms of intelligence is a task remaining to be done, though some fragments are reported in work listed in the bibliography. This work needs to be related to studies of human psychology, neuroscience, biological evolution and comparative ethology.

References

Ballard, D.H. and C.M. Brown, *Computer Vision,* Prentice Hall 1982.

Beaudoin, L.P. and Sloman A, (1993) A study of motive processing and attention, in A.Sloman, D.Hogg, G.Humphreys, D. Partridge, A. Ramsay (eds) *Prospects for Artificial Intelligence,* IOS Press, Amsterdam, pp 229-238, 1993.

Beaudoin, L.P. (1994) *A design-based study of autonomous agents.* PhD thesis, School of Computer Science The University of Birmingham.

Clark, A 'Computation, connectionism and content' in (ed) Yves Kodratoff, *8th European Conference on AI,* Munich, 1988. Dennett, D.C., *Brainstorms* Bradford Books and Harvester Press, 1978.

Dennett, D.C. *The Intentional Stance,* MIT Press/Bradford Books, 1987

Fields, C. 'Measurement and computational description' Proceedings Turing 1990 Colloquium, Sussex University April 1990 (internal report, Knowledge Systems Group, Computing Research Lab, New Mexico State University.)

Fodor, J.A. *Psychosemantics: The Problem of Meaning in the Philosophy of Mind,* MIT Press, Cambridge Mass, 1987.

Funt, B.V. 'Problem-solving with diagrammatic representations' in *Artificial Intelligence,* Vol 13 no 3, pp. 201-230, 1980. Reprinted in R.J. Brachman and H.J. Levesque (eds), *Readings in Knowledge Representation,* Morgan Kaufmann, 1985.

Haugeland, John, *Artificial Intelligence: The Very Idea,* Bradford Books, MIT Press, 1985.

E. Nagel and J.R. Newman *Gödel's Proof* Routledge and Kegan Paul 1958.

Pylyshyn, Zenon W., *Computation and Cognition: Toward a Foundation for Cognitive Science* Bradford Books, MIT Press, 1986

Searle, John, 'Minds Brains and Machines' in *The Behavioural and Brain Sciences,* 1980.

Sloman, A, (1971) Interactions between Philosophy and A.I.: the role of intuition and non-logical reasoning in intelligence, in *Proc. 2nd International Joint Conference on Artificial Intelligence,* London 1971. Reprinted in *Artificial Intelligence,* pp 209-225, 1971, and in J.M.

Nicholas (ed), *Images, Perception, and Knowledge* Dordrecht-Holland: Reidel 1977.

Sloman, A, (1978) *The Computer Revolution in Philosophy: Philosophy Science and Models of Mind,* Harvester Press, and Humanities Press, 1978.

Sloman, A, (1985a) 'Why we need many knowledge representation formalisms', in *Research and Development in Expert Systems,* ed. M Bramer, pp 163-183, Cambridge University Press 1985. Also Cognitive Science Research paper No 52, Sussex University.

Sloman, A, (1985b) 'What enables a machine to understand?' in *Proceedings 9th International Joint Conference on AI,* Los Angeles, 1985.

Sloman, A, Did Searle attack strong strong or weak strong AI, in A.G. Cohn and J.R. Thomas (eds) *Artificial Intelligence and Its Applications,* John Wiley and Sons 1986.

Sloman, A, (1987a) Reference without causal links, in L. Steels, B. du Boulay, D. Hogg (eds), *Advances in Artificial Intelligence-II* (Proc 7th European Conference on AI, Brighton, 1986), 369-381 North-Holland 1987

Sloman, A, (1987b) 'Motives Mechanisms Emotions' in *Emotion and Cognition* 1987, reprinted in M.A. Boden (ed) *The Philosophy of Artificial Intelligence* 'Oxford Readings in Philosophy' Series Oxford University Press, 1990.

Sloman, A, (1989). On designing a visual system: Towards a Gibsonian computational model of vision, in *Journal of Experimental and Theoretical AI* 1,4, 289-337.
(Also available as Cognitive Science Research paper 146, University of Sussex).

Sloman, A. (1992a) 'Prolegomena to a theory of communication and affect' in Ortony, A., Slack, J., and Stock, O. (Eds.) *A.I. and Cognitive Science Perspectives on Communication.* Heidelberg, Germany: Springer, 1992. (Also available as Cognitive Science Research Paper No 194, University of Sussex.)

Sloman, A (1992b) 'The emperor's real mind: review of Roger Penrose's *The Emperor's new Mind: Concerning Computers Minds and the Laws of Physics,'* in *Artificial Intelligence* 56 (1992) pp 355-396
(Also Cognitive Science Research Paper, Birmingham University)

Sloman, A., The mind as a control system, *Philosophy and the Cognitive Sciences, (eds) C. Hookway and D.Peterson,* Cambridge University Press, pp 69-110 1993 (Supplement to *Philosophy*)

Sloman, A, (1994a) Explorations in design space in *Proc ECAI94, 11th European Conference on Artificial Intelligence* Edited by A.G.Cohn, John Wiley, pp 578-582, 1994

Sloman, A, (1994b) Computational modeling of motive-management processes, in Proceedings of the Conference of the International Society for Research in Emotions, Cambridge, July 1994. Ed N.Frijda, p 344-348. ISRE Publications, 1994.

Sloman, A. (1994c), 'Semantics in an intelligent control system,' in *Philosophical Transactions of the Royal Society: Physical Sciences and Engineering* Vol 349, 1689 pp 43-58, 1994

Sloman, A, (1995a) Exploring design space and niche space, in *Proceedings 5th Scandinavian Conf. on AI,* Trondheim May 1995, IOS Press, Amsterdam, 1995

Sloman, A, (1995b) A philosophical encounter: An interactive presentation of some of the key philosophical problems in AI and AI problems in philosophy. *Proc 14th International Joint Conference on AI,* Montreal, 1995.

Sloman, A, (1995c) Towards a general theory of representations, in D.M.Peterson (ed) *Forms of representation* Intellect press (to appear 1995)

Tarski, A, 'The concept of Truth in formalised languages', in *Logic Semantics Metamathematics,* translated by J.H. Woodger, Clarendon Press, Oxford, 1956.

Smith B.C, The semantics of clocks, in James H. Fetzer (ed) *Aspects of Artificial Intelligence,* pp

3-31, Kluwer Academic Publishers, 1988.

C.N.Taylor, *A Formal Logical Analysis of Causal Relations,* D.Phil Thesis, School of Cognitive and Computing Sciences, Sussex University, 1992. (Cognitive Science Research Paper No.257)

Turing, A.M. 'Computing machinery and intelligence' in E.A. Feigenbaum and J. Feldman (eds) *Computers and Thought* McGraw-Hill, New York, 1963, 11-35. (Originally in *MIND* 1950).

Wright, I.P, Sloman, A, & Beaudoin L.P (To appear.) The architectural basis for grief, presented at Geneva Emotions Week 8-13 April 1995.

Young, R.A. 1994, The Mentality of Robots, *Proc. Aristotelian Soc.*