PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

Non-Symbolic Approaches to Artificial Intelligence and the Mind [and Discussion]

David Willshaw, D. Dennett and D. Partridge

Phil. Trans. R. Soc. Lond. A 1994 **349**, 87-102
doi: 10.1098/rsta.1994.0115

Non-symbolic approaches to artificial intelligence and the mind

By David Willshaw

Centre for Cognitive Science, University of Edinburgh, 2 Buccleuch Place, Edinburgh EH8 9LW, U.K.

Current theories of artificial intelligence and the mind are dominated by the notion that thinking involves the manipulation of symbols. The symbols are intended to have a specific semantics in the sense that they represent concepts referring to objects in the external world and they conform to a syntax, being operated on by specific rules.

I describe three alternative, non-symbolic approaches, each with a different emphasis but all using the same underlying computational model. This is a network of interacting computing units, a unit representing a nerve cell to a greater or lesser degree of fidelity in the different approaches. *Computational neuroscience* emphasizes the development and functioning of the nervous system; the approach of *neural networks* examines new algorithms for specific applications in, for example, pattern recognition and classification; according to the *sub-symbolic approach*, concepts are built up of entities called sub-symbols, which are the activities of individual processing units in a neural network.

A frequently debated question is whether theories formulated at the sub-symbolic level are 'mere implementations' of symbolic ones. I describe recent work due to Foster, who proposes that it is valid to view a system at many different levels of description and that, whereas any theory may have many different implementations, in general sub-symbolic theories may not be implementations of symbolic ones.

1. The symbolic approach

The idea that thought involves the manipulation of symbols, the rules of thought being the rules of logic, has a long pedigree, dating back to Hobbes, Descartes, Locke and Hume, among others. The modern variant of this approach has had two major influences. One is the development of the digital computer, which has been seen as a machine for the manipulation of the symbols of formal logic (Haugeland 1981). The second influence, developing through the linguistic theories of Chomsky (1957, 1968) is expressed in the notion that there is a Language of Thought (Fodor 1975). As in linguistic theory, there is a semantics and an interpretable syntax, the set of rules governing the allowable manipulations of symbols representing concepts that refer to objects in the external world.

In a similar vein, most artificial intelligence techniques are based on the application of a set of algorithms to a knowledge base, represented in some formal language. Automated solutions are sought for problems in such areas as game

Phil. Trans. R. Soc. Lond. A (1994) **349**, 87–102
Printed in Great Britain

playing, expert systems, natural language understanding, machine learning, planning, automated reasoning and theorem proving. Approaches to many of these subjects are reviewed elsewhere in this volume.

According to many people, the limited success of such *symbolic* approaches has begged the question whether there are other more suitable techniques. This paper is about *non-symbolic* methods for studying artificial intelligence and the mind. Three approaches are discussed, which although quite different in their objectives, have many family resemblances. The first, *computational neuroscience*, takes the view that, in investigating artificial intelligence and the mind, it is reasonable to examine the only intelligent system known to exist, which involves attempting to understand the central nervous system. The second arose from many years of theorizing about the brain, which led eventually to the emergence of the field of *neural networks*. The preoccupation here is to examine the computational properties of collections of highly interconnected simple computing units, working in parallel, that are modelled loosely on the nerve cells of the brain, and to apply these networks to specific tasks of pattern recognition, classification, and so on. The third approach comes closest to the symbolic approach, and can be viewed as an attempt to substitute neural network based models for symbolic models. Appropriately, this approach is named *sub-symbolic*. Each approach will be described and evaluated.

(*a*) *Criticisms of the symbolic approach*

Many aspects of human reasoning do not seem to follow logical rules. A much cited example is the Wason selection task (Johnson-Laird & Wason 1970). Subjects were presented with a set of cards, each of which had a symbol on each side. Initially they were allowed to see only one side of the cards and were then asked to indicate which cards should be turned over to demonstrate the truth or falsity of a particular proposition (such as 'if there is a vowel on one side, there is an even number on the other'). The response demonstrating that the subjects were applying the rules of symbolic logic was observed in only 4% of cases. Poor logical reasoning in abstract versions of a task and the ability to improve in more real-world versions of it (Wason & Shapiro 1971; Johnson-Laird *et al.* 1972) counters the view that thought is simply the manipulation of logical propositions. A more general criticism is that, for many biological applications, the symbolic approach seems inappropriate. The serial computer analogy that underlies rule-based systems seems inapplicable to the parallel hardware of the central nervous system. Most importantly, it seems difficult to see how many of the tasks that humans perform perfectly, such as those concerned with movement, vision and speech, can be blocked out in terms of the execution of simple rules. As far as artificial intelligence is concerned, Hofstadter (1980) commented 'The strange flavour of AI work is that people try to put together long sets of rules in strict formalisms which tell inflexible machines how to be flexible.'

(*b*) *Levels*

The suggestion that, in some cases, symbolic theories may be inappropriate points towards the issue of the level at which a theory should be formulated and analysed. A widely quoted view is that usually attributed to Marr (1982), according to which there are three levels: the *computational*, at which the nature of the computation to be performed is expressed; the *algorithmic*, at which the

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

procedure for carrying out the computation is formulated; and the *implementational*, the level of hardware. A primary consideration is how many levels there are and whether they are defined arbitrarily or have concrete form. One view (Newell 1982) is that levels are a reflection of the physical world. Working from the bottom, there is the device level, the circuit level, the logic circuit sub-level, the symbol level, and so on. The highest level is the knowledge level, which is akin to the semantic level of Pylyshyn (1984). Dennett (1971) proposes three levels, which he calls stances. The top level, the intentional stance, reflects the ideal computational level and may have little to do with the lower levels. According to Rumelhart & McClelland (1985), there are many algorithmic sub-levels, bounded by the computational level at one extreme and the implementational level at the other.

As we shall see later, one issue has been whether the various levels are independent. It has been argued that symbolic theories occupy one level and other theories are mere implementation (Fodor & Pylyshyn 1988).

2. Computational neuroscience

Computational neuroscience uses mathematical analysis and computer simulation to evaluate theories of the nervous system towards gaining an understanding of the only intelligent system so far known. Compared with the goals of symbolic enterprises, most theories in this field are relatively modest in scope, being concerned with, for example, associative storage and retrieval rather than inductive reasoning; theories for walking rather than chess playing.

Here the issue of levels is important. Churchland & Sejnowski (1992) highlight the fact that in the nervous system there are organized structures on different scales: 'molecules, synapses, neurons, networks, layers, maps and systems'. According to these authors, this range of structural organization implies as many levels of implementation and therefore of algorithms. They suggest that an interlocked set of theories is appropriate for neuroscience, each having a computational, algorithmic and implementational component, but expressed at different levels: a sort of floating triumvirate. Where the triumvirate is anchored depends on the investigator's point of view. For the case of associative memory, for example, to a psychologist the algorithm concerns the manipulation of the information to be stored and the implementation concerns how the strengths of the modifiable synapses underlying storage will be altered; to a physiologist the algorithm concerns how the effect of any synapse is changed under various conditions of depolarisation and excitation of the corresponding neurons, and the implementation is the operation of the voltage sensitive channels of the modifiable synapse; to a pharmacologist the algorithm describes the conditions under which certain molecules cause changes in channel properties, and the implementation is concerned with the underlying chemical reactions.

Theories in computational neuroscience (i) use an entirely different set of concepts from those used in symbolic theories and (ii) are multilevel.

As a first illustration, I continue the example of associative memory. Making use of symbolic and computing analogies will give rise to theories of memory that use such concepts as 'stacks', 'pointers' and 'linked lists', which may be adequate for conventional computer databases but may not necessarily apply to the brain. A more compelling picture, suggested originally by various analogies, such as

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

those aiming to capture the essence of distributed storage (Willshaw *et al.* 1969), is that information is stored in the modifiable synapses at the intersections of the processes of two sets of nerve cells, an organization that is reminiscent of the circuits within the hippocampus (Marr 1971; McNaughton & Morris 1987). Many problems have been addressed at different levels: there is the question of the storage capacity of this general class of architecture (Willshaw *et al.* 1969); the question of the capacity of such networks in which the wiring details such as those found in the hippocampus are incorporated (Willshaw & Buckingham 1990); at a more detailed level is the issue of how the individual computations can be made by the cells (Buckingham & Willshaw 1993); at an even more detailed level is a question, currently of great interest, concerning the intercellular signalling mechanism that enables synaptic modification to take place.

Another class of problems investigated by the techniques of computational neuroscience is concerned with the development of the nervous system. Here again, theories are multilevel and are expressed in terms of differential equations rather than logical formulae. Perhaps the most transparent example is concerned with the development of connections in sensory systems, such as the visual system. It is well known that in all vertebrates there is an ordered projection of the retina onto the main visual centre of the brain (Gaze 1970). In amphibians and fishes, the way in which connections are developed between the ganglion cells of the retinae and the main visual centre, the optic tectum, can be thought of in terms of a process of 'self-organization' of connections: the initial pattern of connections is progressively refined until cells of neighbouring retinal origin come to connect to neighbouring tectal cells. There is then the question of how the self-organizing mechanism is realized, for which a means of interchanging information between cells is required. There are two main candidates, one involving the use of correlated neural activity and one the interchange of diffusible substances (Willshaw & von der Malsburg 1976, 1979). This class of theory is now becoming of increasing importance in the light of evidence that the developing nervous system is shaped by the sensory information that it receives.

As a final example, I look at the type of problem that at one time Artificial Intelligence practitioners would have rejected as trivial but now recognize as being difficult and significant. Like problems such as vision, speech and hand–eye coordination, locomotion is one of those apparently effortless tasks for which satisfactory theories are required. It would seem inappropriate to construct a theory of locomotion in terms of a set of rules, programmed in, for example, Prolog, such as 'If heel is touching floor then move toe downwards', 'If left leg is moving forward then press down on right foot', etc. A theory that is very different from anything built on symbolic principles is due to Taga *et al.* (1991) on the self-organized control of bipedal locomotion. Successful locomotion requires an exquisite interaction between motor system and environment. The suggestion is explored that bipedal locomotion is the result of the entrainment of a set of coupled neural oscillators which drive the muscles and which receive feedback from the environment. The model of locomotion developed is in the form of a set of coupled differential equations representing a set of oscillators. Each oscillator consists of two neurons acting at a joint and inducing torques in opposite directions, which are the directions of contraction of flexor and of extensor muscles. The strengths of the couplings between oscillators are calculated from the known patterns of locomotion observed in humans. For example, in each leg the

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

knee and ankle oscillators produce two sequences of flexion and extension while the hip oscillator is producing just one; the hip unit oscillators of the two legs mutually inhibit each other to give alternation between left and right legs. Simulations of this model reproduce patterns of human walking that are resistant to changes in environmental conditions, such as uphill slopes or rough terrain. A switch from walking to running can be induced through changing the value of a single parameter.

In summary, the theories of computational neuroscience are specific rather than general, are multilevel in character and are expressed in a language whereby, in principle, there is a straightforward mapping of the proposed elements of the theory onto the system that will realize it.

3. Neural networks

As the name suggests, the field of neural networks has its roots in computational neuroscience and has been developing since the 1940s (McCulloch & Pitts 1943; Arbib 1964; Rumelhart *et al.* 1986*a*). It has a number of other names, such as *parallel distributed processing*, *neurocomputing* and *connectionism*. Workers in the field of neural networks are concerned with the development, analysis and application of neurally inspired parallel devices. Such devices have found application in many disciplines, from physics to psychology, engineering to economics, geography to geology.

A neural network is made up of a set of simple computing units which influence each other through modifiable connections, or *weights*. The *activity* of each unit at any moment of time is determined by the combined effect of the activities of the units which influence it, as modulated by the strengths of the appropriate weights. Typically the activity X_i of unit i is computed as

$$X_i = \mathcal{F}(\Sigma_j w_{ij} X_j - \Theta_i),$$

where w_{ij} is the strength of the weight between units i and j and Θ_i is the *threshold* for cell i. $\mathcal{F}$ is the fixed *activation function* which specifies how the accumulated input activation at cell i is transformed into an output signal. $\mathcal{F}$ can have various possible forms, such as $\mathcal{F}(z) = 1$ if $z \geqslant 0$ (i.e. if $\Sigma_j w_{ij} X_j \geqslant \Theta_i$), $\mathcal{F}(z) = 0$ otherwise; or it can be the sigmoidal function $\mathcal{F}(z) = 1/(1 + \mathrm{e}^{-z})$. In both cases $\mathcal{F}(z)$ is a monotonically increasing function of z and varies between 0 and 1.

In the simple feedforward neural network, certain units are designated as *input* units, certain others as *output* units, and units that are neither input nor output units are called *hidden* units (Rumelhart *et al.* 1986*b*). In this case, the task of the network is to compute a given input/output relationship; that is, for each of a given set of values assigned to the input units the network is required to compute specific values to be assigned to the output units. This requires the values of the weights to be set appropriately. An advantage of this type of architecture is that procedures have been developed for progressively changing weight values in response to the errors in the outputs computed by the net as different examples of the input/output pairs to be learned are presented. A learning procedure for networks without hidden units was first developed in the 1950s (Rosenblatt 1958; Block 1962). However, the computational power of these networks is limited, and much of the recent interest has been due to the development of new learning

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

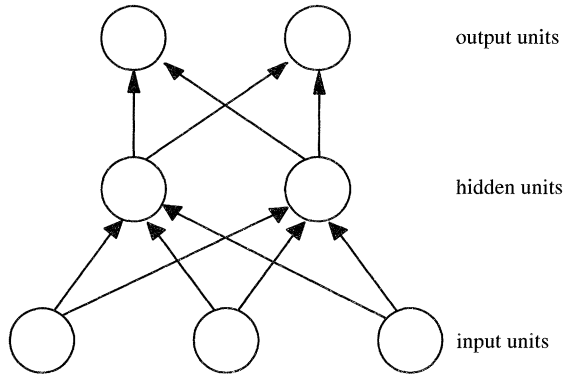

Figure 1. A feedforward neural network with three input units, two hidden units and two output units. Each unit is represented by a circle. The arrows indicate which units are interconnected.

algorithms for more powerful networks that have hidden units (Rumelhart *et al.* 1986*b*).

As a didactic example, I now describe a network that has been trained to learn a simple binary classification problem.

(*a*) *Mirror-symmetry*

As first discussed in the late 1960s (Minsky & Papert 1969), one of the binary classification problems that require hidden units for their solution is that of determining whether a string of 0s and 1s is symmetrical about its mid-point. The computation of *mirror-symmetry* requires just two hidden units, independently of the number of input units (Rumelhart *et al.* 1986*b*).

Figure 2 shows a simple network with weight values set which enables detection of mirror symmetry in strings of 4-bit patterns. The output is 1 if, when the first two bits are reversed, they form the second two bits, and 0 otherwise. The network has four input units, two hidden units and one output unit. The input units are binary valued, and the activities in the other units and the weights are all real valued. The sigmoid activation function is used. It is common practice to treat the threshold Θ associated with each hidden unit and each output unit as if it were an extra weight, called a *bias*, of strength $-\Theta$ on a perpetually active input to the unit in question.

I trained the network of figure 2 by the back propagation learning algorithm (Rumelhart *et al.* 1986*b*). Training was ended when, for all inputs presented, the continuously valued output was within 0.001 of its target value, at which stage the network was deemed to have learned mirror-symmetry. The example of mirror-symmetry will be used later.

(*b*) *Advantages of the neural net approach*

The neural network approach provides a general framework that accommodates a large number of different types of models. There is little doubt that the class of algorithms encompassed within neural networks overlaps those developed in other contexts (such as statistics (Sarle 1994)). However, the range of models covered and the accessibility to the non-specialist has made neural networks attractive to workers from many disparate fields.

Many claims for neural networks over more conventional systems have been

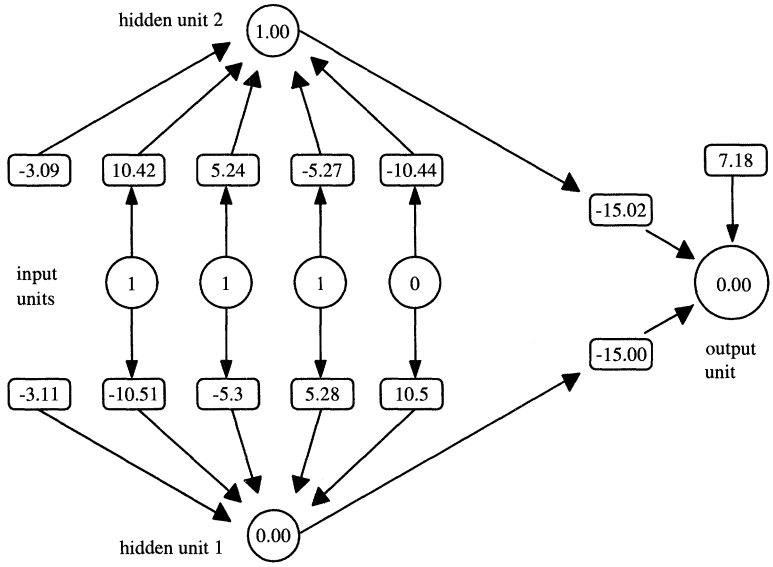

Figure 2. A feedforward neural network for computing mirror symmetry on 4-bit input strings: four input units, two hidden units, one output unit. The numbers within the rectangles represent the strengths of the connections between the corresponding two units, as computed by back propagation; the value of the bias for a unit is shown within a rectangle with no incoming arrow. The numbers within each circle is the value of the appropriate unit's activity for the input string '1 1 1 0'.

made (Smolensky 1987; Clark 1989; Hecht-Nielsen 1990; Hertz *et al.* 1991), which I discuss under the categories *flexibility*, *efficiency of operation*, *brain-like qualities* and *learning*.

1. *Flexibility* covers *graceful degradation* and *generalization*. Graceful degradation (Kohonen 1988) refers to the ability of a network still to function in the face of structural damage. With greater degrees of damage, performance deteriorates gradually rather than catastrophically. This property is a straightforward result of using distributed methods of storage. Generalization concerns the frequently quoted capability of neural networks to respond appropriately to unanticipated inputs, in particular to missing or noisy input data (Clark 1989). There seems to be an element of luck here, since *a priori* it is unlikely that the way the network functions will match the as yet unknown responses to novel stimuli that are required.

2. *Efficiency of operation* covers *numerical*, rather than *logical*, computation, *massively parallel constraint satisfaction* and *fast parallel search*.

It is debatable whether the first two features are unique features of neural networks. The manipulation of real numbers rather than the elements of symbolic logic does not in itself convey any advantage (contrary to the view of fuzzy logic practitioners (Kosko 1992)). Fast parallel search seems to be a positive feature of neural networks, but it is a property of parallel systems in general.

3. *Brain-like qualities*. Neural networks are better models for the nervous system than conventional computers, but are still gross oversimplifications.

4. *Learning*. Much of the rekindled interest in neural networks has been due to the development of new methods for training the weights. On presentation of a

stream of input/output pairs, the weights can be modified gradually in response to the errors made, until the required computation is learned. This obvious advantage has to be tempered with the facts that in some cases the network will learn only provided it has the correct structure; in other cases learning can be slow, or learning on-line may not be an advantage. Notwithstanding, the ability of such systems to learn can be said to represent a real advantage. Learning is possible in certain symbolic systems, but the emphasis on learning is not so strong as in the neural network approach.

4. Neural networks as sub-symbolic systems

A clear exposition of the use of neural networks as sub-symbolic systems was given by Smolensky 1988; see also Rumelhart *et al.* (1986*a*). Both semantics and syntax now have a completely different form from that used in symbolic systems. Each concept is represented by a pattern of activity over a set of units, the firing of each unit representing the presence of a *sub-symbol* (*micro feature*) of the concept. Sub-symbols are not operated on by symbol manipulation but participate in numerical computation; Smolensky (1988) emphasizes that the language of these systems is not that of symbolism but rather that of 'the continuous mathematics describing a dynamical system'. There is now no fixed relation between the representation of an entity and an external object. In a Language of Thought view (Fodor & Pylyshyn 1988), the concepts 'coffee', 'cup', 'sack' and 'jar', for example, each have their own fixed representation. The shades of meaning relating to 'coffee' in a 'cup of coffee', are then conveyed by the co-existence of the symbols for 'cup' and for 'coffee'. In contrast, in a sub-symbolic scheme different patterns of activation represent 'coffee', depending on whether a 'cup of coffee', a 'sack of coffee' or a 'jar of coffee' is meant; context is internal to the representation (Clark 1989). As far as syntax is concerned, the relation between entities are now no longer expressed by rules that resemble those of logic but are represented subtly by the weight values distributed over the network. The process that determines which units are active in a given situation depends delicately on the counterbalance of the effects of the units on each other as modulated by the strengths of the weights between them: 'the operation of soft constraints in a massively parallel fashion'.

(*a*) *Criticisms of the sub-symbolic approach*

It has been argued that many of the advantages claimed for sub-symbolic systems already exist in symbolic systems, particularly those recently developed in Artificial Intelligence. For example, in the original type of production system, which is a set of rules of the 'if ... then' type operating on a database of facts, rules operate one at a time and in serial order (Newell & Simon 1963). However, in more modern systems, the rules act in parallel, and sophisticated methods of assessing priority are used where conflicts arise (Holland *et al.* 1989) ('soft constraints acting in parallel'?). Aside from matters of representation, the principal difference is that in production systems, revision of the rule base involves the explicit revision of the strengths of existing rules or addition of new ones. Sub-symbolic systems provide for rule revision but implicitly through the modification of the existing set of connection strengths; there is no provision for the addition of connections between units that were previously not directly connected.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

Fodor & Pylyshyn (1988) criticize sub-symbolic networks as representing a return to the old associationist ideas of psychology. They argue that symbolic representations do not have the rich structure required of the Language of Thought. As Chater & Oaksford (1990) point out, Fodor & Pylyshyn's strongest claim is that sub-symbolic systems can be best viewed as particular implementations (albeit interesting ones) of symbolic systems. This carries the strong implication that symbolic theories can be formulated and discussed independently of any formulation at a lower level. I now consider another more formal approach which addresses this issue.

5. Algorithms

In recent work, Foster (1992) attempts to interrelate methods for solving a computational task that are defined at different levels by introducing a family of ideal machines with which individual methods can be compared. Each machine represents a description of the behaviour of the physical system on which the given task is being performed, in as much or as little detail as is required. At the highest, most abstract level is the input/output description of the task. At the level of description considered, an *algorithm* is the *sequence of states* through which the system may pass. Each *state* is a list of *values* defined over a set of *labels*, which typically refer to the parameter values taken by the system at a particular step. Foster defines a set of rules for constructing *abstractions* (less detailed descriptions) of a given algorithm. Effectively a family of ideal machines is constructed. Each method can be identified with a specific ideal machine, and methods can be compared by comparing the corresponding ideal machines. Unlike other idealizations (e.g. Turing machines) the descriptions of these ideal machines can be mapped directly onto the underlying physical system.

Two algorithms are said to be *equivalent* if they follow the equivalent set of states. Two states are equivalent if their set of (label, value) pairs are the same, permutations of label being allowed. If algorithm 1 is an abstraction of algorithm 2 then algorithm 2 is said to be an *implementation* of algorithm 1.

(*a*) *Mirror symmetry*

To illustrate use of the formalism, a number of algorithms for detecting mirror symmetry in bit strings are now constructed. The arguments follow closely those of Foster (1992) for the case of exclusive-or.

The first column of table 1 (program A) gives a sample of pseudo-code for detecting whether 4-bit strings are mirror-symmetric. The method used is to check if the value of the first bit matches that of the last bit and that of the second bit matches the penultimate bit. One possible description of the sequence of states through which this program can go is shown in the columns of table 1, for one particular input string. The labels are the set of variables and the next instruction (which has no special status). Each column refers to a particular label, and the entries in that column refer to the values that are associated with the label at each step in the sequence (u means undefined). This is the description for just one input string only; the full description would be for all 16 possible inputs.

Even for this simple task there are other ways of computing mirror-symmetry. One slightly unorthodox way (method B) is based on the observation that if each

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

Table 1. *A description of program A*

	program A	$X1$	$X2$	$X3$	$X4$	$Y1$	$Y2$	Z
1	read $X1$;	u	u	u	u	u	u	u
2	read $X2$;	1	u	u	u	u	u	u
3	read $X3$;	1	1	u	u	u	u	u
4	read $X4$;	1	1	1	u	u	u	u
5	if $X1$=$X4$ then $Y1 = 1$ else $Y1 = 0$;	1	1	1	0	u	u	u
6	if $X2$=$X3$ then $Y2 = 1$ else $Y2 = 0$;	1	1	1	0	0	u	u
7	if $Y1$=1 and $Y2$=1 then Z=1 else Z=0;	1	1	1	0	0	1	u
8	write Z;	1	1	1	0	0	1	0
9	end	1	1	1	0	0	1	0

Table 2. *A description of program B*

	program B	$X1$	$X2$	$X3$	$X4$	Y	Z
1	read $X1$;	u	u	u	u	u	u
2	read $X2$;	1	u	u	u	u	u
3	read $X3$;	1	1	u	u	u	u
4	read $X4$;	1	1	1	u	u	u
5	$Y = 2X1 + X2 - (2X4 + X3)$;	1	1	1	0	u	u
6	if $Y = 0$ then $Z = 1$ else $Z = 0$;	1	1	1	0	2	u
7	write Z;	1	1	1	0	2	0
8	end	1	1	1	0	2	0

half of an n-bit string is interpreted as the binary encoding of a $\frac{1}{2}n$ bit integer (the order of the bits in the second half being reversed), then equality of the two integers will signal mirror symmetry in the whole bit pattern.

For inputs $X1$, $X2$, $X3$, $X4$ we define $Y = 2X1 + X2 - (2X4 + X3)$ and check whether Y equals 0. The Foster description for the input values used in the previous example is shown in table 2.

We can investigate the relationship between algorithms by using the following operations to construct more abstract (less detailed) versions of algorithms.

1. *Selection of states*; selection of some states in the sequence.

2. *Selection of values*; selection of some label-value pairs.

3. *Rounding*; applying some mathematical function to each value to produce a new value.

4. *Duplication*; copying sets of state values.

A trivial abstraction is to choose the entries for $X1$, $X2$, $X3$, $X4$ and Z for the penultimate rows of both table 1 and table 2, and then repeat this operation for each of the 16 possible input strings. This gives two identical descriptions, containing the specification of input and output values only: the highest level

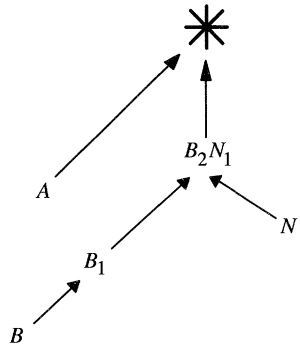

Figure 3. Showing how the algorithms A, B, N and their abstractions for calculating mirror-symmetry are related according to Foster's analysis. Each algorithm is joined by an arrow to the algorithm that is a direct abstraction of it, the arrow pointing towards the more abstract algorithm. The star denotes the most abstract, top-level algorithm, which is the input–output specification.

description of the algorithm for the given input values. This illustrates that there always exists a common abstraction of any two algorithms.

To examine the possible equivalences between algorithms A, B and the neural network method described earlier (called N), I now construct table 3. Each row shows, for a different input string, all (label, value) pairs corresponding to the penultimate lines of tables 1 and 2. For N, the important parameters are the activations of the units of the network, and their values are shown here.

These three descriptions are not equivalent. Program A is described in terms of the binary values for 3 labels; for program B there are just 2 labels with integer values; the network N is described by real values for 3 labels. Table 3 also shows several further abstractions of these algorithms. Firstly, the values describing N are rounded by replacing each number by the nearest integer value, to give N_1. Program B is then abstracted in two stages: first duplicate the values grouped under $Y1$ (with a change of the duplicated label to $Y2$) to form B_1; second, replace all negative values of $Y1$ by 1 and all positive values by 0, and then perform the converse for the values of $Y2$. The result, method B_2, is identical to N_1, once the labelling of the columns has been standardized.

Even for this very simple example, the relations between the various different methods for generating the given set of input/output pairs are not straightforward. The summary description shown in figure 3 demonstrates that none of the algorithms A, B, N is an abstraction (or implementation) of any of the others, and the algorithm which is an abstraction of all three is the trivial (top level) description (marked by a star). The figure also illustrates that one of the symbolic programs, B, is very much closer to the network method, N, than it is to the other symbolic program, A.

In general, as more detail is removed from the possible algorithms at any intermediate level, the algorithms become fewer in number until eventually only the top level description remains. Going in the opposite direction, there is a similar reduction in the number of possibilities, but this is now an effect of the constraints of the physical system becoming increasingly important. There is not necessarily a path between any two algorithms at different levels.

Foster's approach produces a framework for discussing methods for solving a

Table 3. *A comparison of A, B and N*

				A			*B*		*N*		
*X*1	*X*2	*X*3	*X*4	*Y*1	*Y*2	*Z*	*Y*	*Z*	*H*1	*H*2	*O*
0	0	0	0	1	1	1	0	1	0.04	0.04	1.00
0	0	0	1	0	1	0	−2	0	1.00	0.00	0.00
0	0	1	0	1	0	0	−1	0	0.90	0.00	0.00
0	1	0	0	1	0	0	1	0	0.00	0.90	0.00
1	0	0	0	0	1	0	2	0	0.00	1.00	0.00
0	0	1	1	0	0	0	−3	0	1.00	0.00	0.00
0	1	0	1	0	0	0	−1	0	0.89	0.00	0.00
1	0	0	1	1	1	1	0	1	0.04	0.04	1.00
0	1	1	0	1	1	1	0	1	0.04	0.04	1.00
1	0	1	0	0	0	0	1	0	0.00	0.89	0.00
1	1	0	0	0	0	0	3	0	0.00	1.00	0.00
1	1	1	0	0	1	0	2	0	0.00	1.00	0.00
1	1	0	1	1	0	0	1	0	0.00	0.89	0.00
1	0	1	1	1	0	0	−1	0	0.90	0.00	0.00
0	1	1	1	0	1	0	−2	0	1.00	0.04	0.00
1	1	1	1	1	1	1	0	1	0.04	0.04	1.00

B_1			B_2			N_1		
*Y*1	*Y*2	*Z*	*Y*1	*Y*2	*Z*	*H*1	*H*2	*O*
0	0	1	0	0	1	0	0	1
−2	−2	0	1	0	0	1	0	0
−1	−1	0	1	0	0	1	0	0
1	1	0	0	1	0	0	1	0
2	2	0	0	1	0	0	1	0
−3	−3	0	1	0	0	1	0	0
−1	−1	0	1	0	0	1	0	0
0	0	1	0	0	1	0	0	1
0	0	1	0	0	1	0	0	1
1	1	0	0	1	0	0	1	0
3	3	0	0	1	0	0	1	0
2	2	0	0	1	0	0	1	0
1	1	0	0	1	0	0	1	0
−1	−1	0	1	0	0	1	0	0
−2	−2	0	1	0	0	1	0	0
0	0	1	0	0	1	0	0	1

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

given computational task in terms of a family of interrelated ideal machines. The arbitrary specification of the number of levels used by some authors is removed by considering all levels to be levels of description, in more or less detail, of the physical system on which the given task is to be executed. The notion of implementation is couched in terms of the level of detail required. With reference to neuroscience, this would seem to be a more satisfactory approach than reinterpreting the notions of 'computation', 'algorithm' and 'implementation' according to the level being considered. The general picture is that sub-symbolic algorithms represent more detailed descriptions than symbolic ones, by virtue of the fact that their usual level of description involves specification of the activities of units as real numbers. Symbolic algorithms do not occupy a privileged position, but represent just one possible level of description. The dividing lines between these two types of algorithms cannot be drawn easily: a symbolic algorithm may resemble a sub-symbolic algorithm more closely than it does another symbolic one. With respect to the usual claim that connectionist algorithms are mere implementations, it cannot be said of a (symbolic) algorithm that there are so many implementations of it that there is nothing special about any particular (sub-symbolic) one. Any algorithm may have many different implementations, although what implementations of a given algorithm are valid are constrained by the nature of the physical system on which the algorithm runs and for which all algorithms are approximations, in a greater or a lesser degree of detail. Other constraints are supplied by causality and interpretation, which Foster deliberately does not consider.

6. Conclusions

At present we have little insight into how to program a machine to analyse a visual scene, or navigate in a real-world environment, let alone display any sort of 'higher-level' abilities. The symbolic approach to cognition and artificial intelligence has been pursued with only limited success. It would be well to recognize that this is just one of a number of different approaches. Three alternative non-symbolic approaches have been reviewed. The nature of each of them is conditioned by the way in which they have developed and the type of problem that they address. Whereas non-symbolic and symbolic approaches embody equivalent computational power, they display significant differences. Non-symbolic approaches tend to imply inherently parallel and more brain-like theories which are interpretable at many different levels. Non-symbolic theories tend to be relatively modest in scope but are inherently flexible, incorporating mechanisms that are adaptable through learning. They are usually formulated at a more detailed level than symbolic ones. However, it does not follow that the former type is merely an implementation of the latter, which is a matter for investigation in each particular case. The level at which any theory should be formulated depends on the questions that are being asked, and the symbolic level is just one among many.

I thank Carol Foster, Bruce Graham and Peter Dayan for commenting on this manuscript and the Medical Research Council for financial support under Programme Grant PG9119632.

References

Arbib, M. A. 1964 *Brains, machines, and mathematics.* New York: McGraw-Hill.

Block, H. D. 1962 The perceptron: A model for brain functioning. *Rev. Mod. Phys.* **43**, 123–135.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

Buckingham, J. T. & Willshaw, D. J. 1993 On setting unit thresholds in an incompletely connected associative net. *Network* **4**, 441–459.

Chater, N. & Oaksford, M. 1990 Autonomy, implementation and cognitive architecture: A reply to Fodor and Pylyshyn. *Cognition* **34**, 93–107.

Chomsky, N. 1957 *Syntactic structures.* The Hague: Mouton.

Chomsky, N. 1968 *Language and mind.* New York: Harcourt, Brace and World.

Churchland, P. S. & Sejnowski, T. J. 1992 *The computational brain.* Cambridge, Mass: MIT Press, Bradford Books.

Clark, A. 1989 *Microcognition: philosophy, cognitive science and parallel distributed processing.* Cambridge, Mass.: MIT Press/Bradford Books.

Dennett, D. C. 1971 Intentional systems. *J. Philos.* **68**, 87–106.

Fodor, J. A. 1975 *The language of thought.* New York: Crowell.

Fodor, J. A. & Pylyshyn, Z. W. 1988 Connectionism and cognitive architecture: A critical analysis. *Cognition* **28**, 3–71.

Foster, C. L. 1992 *Algorithms, abstraction and implementation.* London: Academic Press.

Gaze, R. M. 1970 *The formation of nerve connections.* London: Academic Press.

Haugeland, J. 1981 Semantic engines: an introduction to mind design. In *Mind design* (ed. J. Hagueland). Montgomery, Vermont: Bradford Books.

Hecht-Nielsen, R. 1990 *Neurocomputing.* Addison-Wesley.

Hertz, J., Krogh, A. & Palmer, R. G. 1991 *Introduction to the theory of neural computation.* Addison-Wesley.

Hofstadter, D. R. 1980 *Godel, Escher, Bach: an eternal golden braid.* Penguin Books.

Holland, J. H., Holyoak, K. J., Nisbett, R. E. & Thagard, P. R. 1989 *Induction: processes of inference, learning and discovery.* Cambridge, MA: MIT Press.

Johnson-Laird, P. N. & Wason, P. C. 1970 A theoretical analysis of insight into a reasoning task. *Cognitive Psychology* **10**, 64–99.

Johnson-Laird, P. N., Legrenzi, P. & Legrenzi, M. S. 1972 Reasoning and a sense of reality. *Br. J. Psychol.* **63**, 395–400.

Kohonen, T. 1988 *Self organisation and associative memory.* Berlin: Springer-Verlag.

Kosko, B. 1992 *Neural networks and fuzzy systems.* Englewood Cliffs, New Jersey: Prentice-Hall.

Marr, D. 1971 Simple memory: A theory for archicortex. *Phil. Trans. R. Soc. Lond.* B**262**, 23–81.

Marr, D. 1982 *Vision.* San Francisco: Freeman.

McCulloch, W. S. & Pitts, W. H. 1943 A logical calculus of ideas imminent in nervous activity. *Bull. Math. Biophys.* **5**, 115–133.

McNaughton, B. L. & Morris, R. G. M. 1987 Hippocampal synaptic enhancement and information storage within a distributed memory system. *Trends in Neurosci.* **10**, 408–415.

Minsky, M. A. & Papert, S. 1969 *Perceptrons.* Cambridge, Mass.: MIT Press.

Newell, A. 1982 The knowledge level. *Artificial Intelligence* **18**, 87–127.

Newell, A. & Simon, H. 1963 GPS: A program that simulates human thought. In *Computers and thought* (ed. E. A. Feigenbaum & J. Feldman). New York: McGraw Hill.

Pylyshyn, Z. W. 1984 *Computation and cognition.* Cambridge, Mass.: MIT Press.

Rosenblatt, F. 1958 The perceptron: A probabilistic model for information storage and organisation in the brain. *Psychol. Rev.* **65**, 368–408.

Rumelhart, D. E. & McClelland, J. L. 1985 Levels indeed! A response to Broadbent. *J. Experimental Psychol. General* **114**, 193–197.

Rumelhart, D. E., Hinton, G. E. & McClelland, J. L. 1986*a* A general framework for parallel distributed processing. In *Parallel distributed processing: explorations in the microstructure of cognition. Vol. I: Foundations* (ed. D. E. Rumelhart & J. L. McClelland). Cambridge, Mass.: MIT Press.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. 1986*b* Learning representations by back-propagating errors. *Nature, Lond.* **323**, 533–536.

Sarle, W. S. 1994 Neural networks and statistical models. In *Proceedings of the 19th Annual SAS User Group International Conference.*

Smolensky, P. 1987 The constituent structure of connectionist mental states: A reply to Fodor and Pylyshyn. *Southern J. Philos.* **26**, 137–159.

Smolensky, P. 1988 On the proper treatment of connectionism. *Behavioural Brain Sci.* **11**, 1–74.

Taga, G., Yamaguchi, Y. & Shimizu, H. 1991 Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics* **65**, 147–159.

Wason, P. C. & Shapiro, D. 1971 Natural and contrived experience in a reasoning problem. *Q. Jl Experimental Psychol. Human Experimental Psychol.* **23**, 63–71.

Willshaw, D. J. & Buckingham, J. T. 1990 An assessment of Marr's theory of the hippocampus as a temporary memory store. *Phil. Trans. R. Soc. Lond.* B**329**, 205–215.

Willshaw, D. J. & von der Malsburg, C. 1976 How patterned neural connexions can be set up by self-organisation. *Proc. R. Soc. Lond.* B**194**, 431–445.

Willshaw, D. J. & von der Malsburg, C. 1979 A marker induction mechanism for the establishment of ordered neural mappings: its application to the retinotectal problem. *Phil. Trans. R. Soc. Lond.* B**287**, 203–243.

Willshaw, D. J., Buneman, O. P. & Longuet-Higgins, H. C. 1969 Non-holographic associative memory. *Nature, Lond.* **222**, 960–962.

Discussion

D. Dennett (*Tufts University, U.S.A.*). The result that two algorithms are equivalent according to a specific set of abstract rules is not very interesting. They are still two different ways of solving the problem.

D. Willshaw. True. However, the question concerns the relationship of relative abstraction that exists between two different algorithms which, at the highest level, effect the same input–output mapping. We must also ask which algorithms can be implemented in neurobiology.

D. Partridge (*University of Exeter, U.K.*). Are the abstractions that transform a sub-symbolic algorithm into a representation identical to that obtained from a symbolic algorithm? And is there any limitation on what transformation abstractions are allowed?

D. Willshaw. Many abstraction operations are possible but Foster considered a particular set. Some of these seem intuitively reasonable; for some others (such as rounding), it is obviously important to define their scope, and more work is needed on that point.

D. Partridge. Then I wonder what this work, as it currently stands, really shows. For if any abstraction is allowed, then any such representation can be transformed into any other (I suspect). It's just a matter of devising the appropriate 'abstraction'. Now, a question for both Dr Willshaw and Professor Brady. They both stress that by building 'intelligent' robots interacting with the real world, we are forced to tackle the tricky problems (e.g. noise) that purely software models usually avoid. And they both implied that intelligence is an embodied phenomenon, so robotics is more appropriate than the more popular abstract simulations. But the only intelligent systems we know of are embodied in the

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

'wetware' of the human brain. So why should electronic robots yield more insight into human intelligence than disembodied abstractions will?

M. Brady (*University of Oxford, U.K.*). A good try! But the problem is that, at present, we can't build anything like human-brain architectures to run our robotics research.

D. Partridge. I wasn't suggesting that we should build brain-like machines, but that traditional AI isn't obviously inferior to robotics. That AI has ignored various problematic aspects of the real world is true, and perhaps it shouldn't always have done so. But one can't rule out abstraction as a methodology. Scientists always abstract from reality to make progress. The central question is the true significance of what is ignored, and whether it can be added back in later, if required.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES