HOW LOGIC EMERGES FROM THE DYNAMICS OF INFORMATION

Peter Gärdenfors

Lund University Cognitive Science
Kungshuset, Lundagård
S–223 50 Lund
Sweden
E-mail: Peter.Gardenfors@fil.lu.se

Abstract: It is often claimed that the symbolic approach to information processing is incompatible with connectionism and other associationist modes of representing information. I propose to throw new light on this debate by presenting two examples of how logic can be seen as emerging from an underlying information dynamics. The first example shows how intuitionistic logic results very naturally from an abstract analysis of the dynamics of information. The second example establishes that the activities of a large class of neural networks may be interpreted, on the symbolic level, as nonmonotonic inferences. On the basis of these examples I argue that symbolic and non-symbolic approaches to information can be described in terms of different perspectives on the same phenomenon. Thus, I find that Fodor and Pylyshyn's claim that connectionist systems cannot be systematic and compositional is based on a misleading interpretation of representations in such systems.

1. TWO PARADIGMS OF COGNITIVE SCIENCE

There are currently two dominating paradigms concerning how cognitive processes can be identified. The first is the *symbolic* paradigm according to which the atoms of mental representations are symbols which combine to form meaningful expressions. Information processing involves above all computations of logical consequences. In brief, the mind is seen as a Turing machine that operates on sentences from a mental language by symbol manipulation. The second paradigm claims that cognition is characterized by *associations*. This idea goes back to the empiricist philosophers, but it has recently seen a revival in the emergence of *connectionism*. It has been often been claimed that these two paradigms are fundamentally at odds with one another, most notably by Fodor and Pylyshyn (1988). I shall argue that they are not.

1.1 The symbolic paradigm

The central tenet of the symbolic paradigm is that mental representation and information processing is essentially *symbol manipulation*. The symbols can be concatenated to form expressions in a *language of thought* – sometimes called Mentalese. A *mental state* is identified with a set of attitudes towards such expressions.

The content of a sentence in Mentalese is a proposition or a thought of a person. The different propositional attitudes in the mental states of a person are connected via their *logical* or *inferential relations*. Pylyshyn writes (1984, p. 194): “If a person believes (wants, fears) P, then that person's behavior depends on the form the expression of P takes rather than the state of affairs P refers to ... .” In applications within AI, first order logic has been the dominating inferential system, but in other areas more general forms of inference, like those provided by statistical inference, inductive logic or decision theory, have been utilized.

Processing the information contained in a mental state consists in computing the consequences of the propositional attitudes, using some set of *inference rules*. The following quotation from Fodor (1981, p. 230) is a typical formulation of the symbolic paradigm:

> ”Insofar as we think of mental processes as computational (hence as formal operations defined on representations), it will be natural to take the mind to be, inter alia, a kind of computer. That is, we will think of the mind as carrying out whatever symbol manipulations are constitutive of the hypothesized

> computational processes. To a first approximation, we may thus construe mental operations as pretty directly analogous to those of a Turing machine."

The material basis for these processes is irrelevant to the description of their results – the same mental state with all its propositional attitudes can be realized in a brain as well as in a computer. Thus, the symbolic paradigm clearly presupposes a *functionalist* philosophy of mind. The inference rules of logic and the electronic devices which conform to these rules are seen to be analogous to the workings of the brain. In brief, the mind is thought to be a computing device, which generates symbolic sentences as inputs from sensory channels, performs logical operations on these sentences, and then transforms them into linguistic or non-linguistic output behaviors.

A further claim of the symbolic paradigm is that mental representations *cannot be reduced* to neurobiological categories. The reason is that the functional role of the symbolic representations and the inference rules can be given many different realizations, neurophysiological or others. The causal relations governing such a material realization of a mental state will be different for different realizations, even if they represent the same logical relations. Thus, according to functionalism, the logical relations that characterize mental representations and the information processing cannot be reduced to any underlying neurological or electronic causes. (Cf. P. S. Churchland 1986, Ch. 9.5.)

The outline of the symbolic paradigm that has been presented here will not be explicitly found in the works of any particular author. However, a defense of the general reasoning can be found, for example, in the writings of Jerry Fodor (1981, Introduction, and Chs. 7 and 9) and Zenon Pylyshyn (1984), and in their joint article Fodor and Pylyshyn (1988). It is also clear that the symbolic paradigm forms an implicit methodology for most of the research in AI.

1.2 The associationist paradigm

For Hume, thinking consists basically in the forming of *associations* between "perceptions of the mind." This idea has since then been developed by the British empiricists, the American pragmatists (William James), and, in particular, by the behaviorists. Their stimulus–response pairs are prime examples of the notion of an association. Dellarosa (1988, p. 29) summarizes the central tenet as follows:

> ”Events that co-occur in space or time become connected in the mind. Events that share meaning or physical similarity become associated in the mind. Activation of one unit activates others to which it is linked, the degree of activation depending on the strength of association. This approach held great intuitive appeal for investigators of the mind because it seems to capture the flavor of cognitive behaviors: When thinking, reasoning, or musing, one thought reminds us of others."

During the last decades, associationism has been revived with the aid of a new model of cognition: *connectionism*. Connectionist systems, also called neural networks, consist of large numbers of simple but highly interconnected units (”neurons”). Each unit receives activity, both excitatory and inhibitory, as input; and transmits activity to other units according to some function (normally non-linear) of the inputs. The behavior of the network as a whole is determined by the initial state of activation and the connections between the units. The inputs to the network also change the ’weights’ of the connections between units according to some learning rule. Typically, the change of connections is much slower than changes in activity values. The units have no memory of themselves, but earlier inputs may be represented indirectly via the changes in weights they have caused.[1] In the literature one finds several different kinds of connectionist models (Rumelhart and McClelland 1986, Beale and Jackson 1990, Zornetzer, Davis and Lau 1990) that can be classified according to their architecture or their learning rules.

Connectionist systems have become popular among psychologists and cognitive scientists since they seem to be excellent tools for building models of associationist theories. And networks have been developed for many different kinds of tasks, including vision, language processing, concept formation, inference, and motor control. (Beale and Jackson 1990, Zornetzer, Davis and Lau 1990). Among the applications, one finds several that traditionally were thought to be typical symbol processing tasks. In favor of the neural networks, it is claimed by the connectionists that these models do not suffer from the brittleness of the symbolic models and that they are much less sensitive to noise in the input (Rumelhart and McClelland 1986).

1.3 The unification program: Different perspectives on information

It is generally claimed that the symbolic and the associationist/connectionist paradigms are *incompatible*. Some of the most explicit arguments for this position have been put forward by Smolensky (1988, p. 7) and Fodor and Pylyshyn (1988). Smolensky argues that, on the one hand, symbolic programs requires linguistically formalized precise rules that are sequentially interpreted (hypothesis 4a

[1] For a more formal treatment of neural networks, see Section 3.1.

in his paper); and, on the other hand, connectionist systems cannot be given a complete and formal description on the symbolic level (hypothesis 8).

He also rebuts the argument that, in principle, one type of system can be *simulated* by a system of the other kind. Firstly, he argues, connectionist models cannot be “merely implementations, for a certain kind of parallel hardware, of symbolic programs that provide exact and complete accounts of behavior at the conceptual level” (Hypothesis 10, p. 7) since this conflicts with the connectionist assumption that neural networks cannot be completely described on the symbolic (”conceptual”) level (Hypothesis 8c, pp. 6–7). Secondly, even if a symbolic system is often used to implement a connectionist system, “the symbols in such programs represent the activation values of units and the strength of connections” (p. 7), and they do not have the conceptual semantics required by the symbolic paradigm. Thus the translated programs are not symbolic programs of the right kind.

Fodor and Pylyshyn’s (1988) main argument for the incompatibility of the symbolic and the associationist/connectionist paradigms is that connectionist models, in contrast to the symbolic, lack *systematicity* and *compositionality*.[2] By saying that the capabilities of a system are systematic they mean that “the ability to produce/understand some sentences [symbolic expressions] is *intrinsically* connected to the ability to produce/understand certain others.” (*ibid.*, p. 37) To give a simple example, if you can express or represent “Abel hits Cain” and “Beatrice kisses David,” you can also express or represent, e.g., “Cain hits David” and “Abel kisses Beatrice.” Compositionality is a well-known principle which requires that symbolic expressions can be composed into new meaningful expressions. This principle is required to “explain how a finitely representable language can contain infinitely many nonsynonymous expressions.” (*ibid.*, p. 43) Fodor and Pylyshyn’s arguments for why connectionist systems cannot be systematic and compositional will be presented (and criticized) in Section 4.2.

It thus seems that there is an impenetrable wall between the symbolic and the connectionist paradigms. One of my aims in this article is to show that they can be unified. I will argue that the alleged conflict between these paradigms can be resolved by adopting two different *perspectives* on how information is processed in various systems.

One perspective on an information processing system is to look at its *dynamics*, i.e., how one state of the system is transformed to another, given a particular input to the system. This perspective is the normal one to use when describing a connectionist system. The other perspective is to forget about the details of the transition from the input to the output and only consider what is represented by the input and its relation to what is represented by the output. As will be shown below, this relation can often be interpreted as a symbolic *inference*, completely in accordance with the requirements of the symbolic paradigm. Thus, in a sense to be made more precise later, by changing from one perspective to the other, symbolic inferences can be seen as *emerging* from dynamic ’associations’. The pivotal point is that *there is no need to distinguish between two kinds of systems* – the two perspectives can be adopted on a single information processing system.

I will start out, in Section 2, by describing the dynamics of an information processing system in a very abstract way. Here the details of the process are of no importance. What counts is merely the relation between the input and the output. Using this simple structure, I shall introduce a definition of what a *proposition* is, which does not presume any form of symbolic structure. Nevertheless, if one looks upon these propositions from another perspective, it turns out that there is a *logic* to them.

In Section 3, I will become more concrete and actually use connectionist systems as models of the dynamic processes. Again, by adopting a different perspective on what the system is doing, I shall show that it can be seen as performing logical inferences. It turns out that nonmonotonic inferences can be modelled in a natural fashion by such systems.

In Section 4, I shall return to the alleged clash between the symbolic and the connectionist paradigms. In the light of the example from Section 3, I shall argue that Fodor and Pylyshyn’s criticism of connectionist systems is misplaced. Furthermore, I shall use the examples from Sections 2 and 3 to support the claim that there is no fundamental conflict between the two views.

2. THE DYNAMICS OF INFORMATION AS A BASIS FOR LOGIC

The proper objects of logic are not sentences but the *content* of sentences. Thus, in order to understand what logic is about, one needs a theory of propositions. In traditional philosophical logic, a proposition is often defined in terms of possible worlds, so that a proposition is identified with the set of worlds in which it is true.

With this definition, it is easy to see how the *logic* of propositions can be generated. By using standard set-theoretical operations, we can form composite propositions: The conjunction of two propositions is represented by the intersection of the sets of possible

[2] They also claim that they lack *productivity* (Section 3.1 in their paper) and *inferential coherence* (Section 3.4), but these arguments seem to carry less weight for them.

worlds representing the propositions; the disjunction is represented by their union; the negation is represented by the complement with respect to the set of all possible worlds; etc. As is easily seen, this way of constructing the standard logical connectives results in *classical* truth-functional logic. The underlying reason is simply that the 'logic' of the set-theoretical operations is classical. In this sense we see how already the *ontology* used when defining propositions determines their logic.

2.1 An alternative definition of propositions

I shall now present another way of defining a proposition, based on the *dynamics of information states*, and show how this definition leads to a different perspective on logic. The construction presented here is adapted from Gärdenfors (1984, 1985).

The ontologically fundamental entities in the reconstruction of a propositional logic will be *information states*.[3] In this section, no assumptions whatsoever will be made about the structure of the information states. However, the interpretation is that they represent states in *equilibria* in the sense that the underlying dynamic processes are assumed to have operated until a stable state is reached.[4]

What can change an information state in equilibrium is that it receives some form of informational input[5] that upsets the equilibrium and starts the dynamic processes again. Here, I will avoid all problems connected with a more precise description of *what* an informational input is. I will simply identify an input with the change it induces in an information state.

Formally, this idea can be expressed by defining an informational input as a *function* from information states to information states. When a function representing a certain input is applied to a given information state *K*, the value of the function is the state which would be the result of accommodating the input to *K*. This way of defining informational input via changes of belief is analogous to defining events via changes of physical states.

If two inputs always produce the same new information state, i.e., if the inputs are identical as functions, there is no epistemological reason to distinguish them. Apart from information states, the only entities that will be assumed as primitives in this section are functions of this kind which take information states as arguments and values.

The most important type of input is when new evidence is accepted as certain or 'known' in the resulting information state. Below, I will concentrate on this type of input.[6] Input corresponding to accepting evidence as certain represents the simplest kind of expanding an information state and is one way of modelling learning. The information contained in such an input will be called a *proposition*. Following the general identification of inputs presented above, *propositions are defined as functions from information states to information states*. This definition will be the point of departure for the reconstruction of propositional logic from the dynamics of information. Veltman (1991, p.1) gives an informal account of this definition in the following way: "You know the meaning of a sentence if you know the change it brings about in the information state of anyone who accepts the news conveyed by it."

As a first application of the definition, a central concept for information can now be introduced: A proposition *A* is said to be *accepted as known in the information state K* if and only if $A(K) = K$.

It is important to keep in mind that not all functions that can be defined on information states are propositions. Propositions correspond to a certain type of informational input, to wit, when new evidence is accepted as certain. In order to characterize the class of propositions, I will next formulate some postulates for propositions. Before this is done, we cannot speak of the *logic* of propositions.

2.2 Basic postulates for propositions

First we need a definition of the basic dynamic structure. A *dynamic model*[7] is a pair <K, P>, where K is a set and P is a class of functions from K to K. Members of K will be called information states and they will be denoted *K*, *K'*, … . The elements in P represent the propositions. *A*, *B*, *C*, ... will be used as variables over P. It should be noted once again that nothing is assumed about the structure of the elements in K.

It will be assumed that the informational inputs corresponding to propositions can be iterated and that the composition of two such inputs is also a proposition. The composition of two propositions *A* and *B* will be denoted *A*3*B* (remember that *A*3*B* is not an element of some formal language, but a function from information states to information states). Formally, this requirement is expressed in the following postulate:

[3] Information states were called states of belief in Gärdenfors (1984) and (1988).

[4] Cf. the 'resonant states' described in Section 3.1.

[5] Informational inputs were called epistemic inputs in Gärdenfors (1984) and (1988).

[6] Other forms of informational inputs are discussed in Gärdenfors (1988).

[7] Dynamic models were called belief models in Gärdenfors (1988).

> ($P1$) For every A and B in P, there is a function $A3B$ which is also in P such that, for every K in K, $A3B(K) = A(B(K))$.

It will also be postulated that the composition operation is commutative and idempotent:[8]

> ($P2$) For every A and B in P and for every K in K, $A3B(\mathrm{K}) = B3A(K)$.

> ($P3$) For every A in P and for every K in K, $A3A(K) = A(K)$.

We can now introduce the fundamental relation of *logical consequence* between propositions: A proposition B is a consequence of a proposition A in a dynamic model <K, P>, if and only if $B(A(K)) = A(K)$, for all K in K.

The identity function, here denoted by Å, will be assumed to be a proposition:

> ($P4$) The function Å, defined by $\text{Å}(K) = K$, for all K in K, is in P.

A proposition A is a *tautology* in the dynamic model <K, P>, if and only if $A(K) = \text{Å}(K)$, for all K in K.

The next postulate will be a formal characterisation of the information obtained when one learns that one thing *implies* another (or is *equivalent* to another).

> ($P5$) For every A and B in P, there is a function C in P such that
>
> (a) for all K in K, $A(C(K)) = B(C(K))$;
>
> (b) for any function D in P such that $A(D(K)) = B(D(K))$ for all K in K, there is a function E in P such that $D(K) = E(C(K))$, for all K in K.

A function C which satisfies (a) and (b) will be called an *equalizer* of A and B. With the aid of ($P2$) it can be shown that there is only one equalizer of A and B in P. We can thus give the proposition postulated in ($P5$) a well defined name: the equalizer of A and B will be denoted $A \times B$. From this we define the proposition $A \varnothing B$ which corresponds to the information that A implies B, by the equation $(A \varnothing B)(K) = (A \times (A3B))(K)$. for all K in K.

The negation of a proposition will be defined by first assuming the existence of a 'falsity' proposition:

> ($P6$) In every dynamic model <K, P> there exists a constant function $\perp$ in P, i.e., there is some $K_\perp$ in K such that $\perp(K) = K_\perp$ for all K in K.

$K_\perp$ will be called the *absurd* information state. As is standard in propositional logic, the *negation* $\neg A$ of a proposition A is defined as the proposition $A \varnothing \perp$.

The postulate for disjunction will be in the same style as the postulate concerning equalizers:

> ($P7$) For every A and B in P, there is a function C in P such that
>
> (a) for all K in K, $A(C(K)) = A(K)$ and $B(C(K)) = B(K)$;
>
> (b) for any function D in P that satisfies (a), there is a function E in P such that $E(D(K)) = C(K)$ for all K in K.

A function C that satisfies (a) and (b) will be called a *disjunction* of A and B.

2.3 Completeness results

I have now introduced postulates for operations on propositions that correspond to each of the standard propositional connectives. I will next present some technical results which answer the question of which 'logic' is determined by these postulates.

The crucial point in my construction is that *expressions like* $(A \times (A3B))(K)$ *can be viewed from two perspectives*. Officially, the expressions like $A \times (A3B)$ are not sentences in a language but *functions* defined on information states. However, given that the postulates ($P1$) – ($P7$) are satisfied for the class of functions in P there is, of course, an obvious one–one correspondence between the propositions in a dynamic model and the sentences in a standard 'propositional' language (Å and $\perp$ are sentential constants in this language). In other words, when ($P1$) – ($P7$) are satisfied, a syntactic structure 'emerges' from the class of functions. This entails that we can consider the propositions that are tautologies in a given dynamic model as a class of sentences and then ask how the formulas which are included in all such classes can be axiomatized. If we really want to have an explicit symbolic structure, we can view expressions of the form $A \times (A3B)$ as *names* of the functions. The point is that the referents of the names are uniquely determined by the names themselves.[9] In other words, the *semantics* of the language is trivial: it is the identity mapping. However, in this mapping the same object is given a double interpretation: on the one hand it is a symbolic expression in a formal language; on the other, it is a function in a dynamic model.

[8] For motivations of these and the following postulates, see Gärdenfors (1984).

[9] It is interesting to note that the key idea behind a Henkin completeness proof is based on the same kind of identification: The objects in the Henkin models are determined from equivalence classes of formulas.

It can be shown (Gärdenfors 1985) that the logic generated by the postulates (*P1*) – (*P7*) is *exactly* intuitionistic logic.[10] In order to derive classical propositional logic, we need one more postulate for the class of propositions in a dynamic model:

(*P8*) For every A and B in P, $A\times B = \neg A\times\neg B$.

In this section, I have shown how propositional logic can be constructed from informational dynamics. The key idea for the construction is the definition of a proposition as a function representing changes of belief. The ontological basis of the construction is very meagre. The only entities that have been assumed to exist are information states and functions defined on information states. In order to emphasize this further, let me mention some things that have *not* been assumed: Firstly, it is not necessary that there be an independent object language that expresses the propositions to be studied. In contrast, the structure of an appropriate language emerges from the class of functions when the postulates are satisfied. Secondly, no set theory has been used; all constructions have been expressed solely in terms of functions. Thirdly, it can be noted that the construction does not use the concept of truth or possible worlds in any way.

The dynamic approach to logic presented in this section has been generalized in a several ways by a number of logicians in Amsterdam. Veltman (1991) extends it to an analysis of the function of 'might' and to default rules in general. Groenendijk and Stokhof (1991) provide a dynamic interpretation of first-order predicate logic. Their dynamic predicate logic can be seen as a compositional, non-representational discourse semantics. Apart from giving a dynamic analysis of quantifiers, they show in particular how this approach can be used to handle anaphoric reference. They also compare it to Kamp's (1981) discourse representation theory. In Groenendijk and Stokhof (1990), they extend their approach to a typed language with λ-abstraction and use it to supply a semantic component for a Montague-style grammar. Van Eijck and de Vries (1992) use the approach of Groenendijk and Stokhof and extend dynamic predicate logic with ι-assignments and with generalized quantifiers. Again, their semantics is applied to problems of anaphoric reference. Van Benthem (1992) adopts a very general approach and discusses a number of ways of connecting a dynamic approach to logic with other more traditional logical and algebraic theories. A special case of this is the dynamic modal logic DML developed in de Rijke (1993).

3. Neural networks as nonmonotonic inference machines

In physical systems one often finds descriptions of 'slow' and 'fast' aspects of dynamic processes. A well-known example from statistical mechanics is the 'slow' changes of temperature as a different perspective on a complex system of 'fast' moving gas molecules. Another example is catastrophe theory (Thom 1972) which is an entire mathematical discipline devoted to investigating the qualitative properties (in particular the 'catastrophes') of the 'slow' manifolds generated by a dynamical system.

The analogy I want to make in the context of the conflict between the symbolic and the associationist paradigms is that associationism deals with the 'fast' behavior of a dynamic system, while the symbolic structures may emerge as 'slow' features of such a system. In particular, inferential relations can be described from both perspectives. The upshot is that one and the same system, depending on the perspective adopted, can be seen as both an associationist mechanism and as an inferential rule-following process operating on symbolic structures.

In this section I shall elaborate on this double interpretation for the case when the system is a *neural network*.[11] Pictorially, the 'fast' behavior of a neural network are the 'associations' between the neurons in the network, i.e., the transmission of the activity levels within the network. In other words, what the network does is to locate minima in a 'cognitive dissonance function' (which, e.g., can be identified as maxima in Smolensky's (1986) harmony functions). I want to argue that the corresponding 'slow' behavior of many networks can be described as the results of the network performing *inferences* in a precisely defined sense, and with a well-defined logical structure. It turns out the these inferences are, in a very natural way, *nonmonotonic*.

It should be emphasized that there is a different, even slower, process in a neural network, namely the *learning* that occurs from new instances being presented to the system and which causes the connections between the neurons to change. As is argued in Gärdenfors (1992), this kind of change within a neural network corresponds to another kind of inference, to wit, *inductive* inferences. In this

[10] This follows essentially from the fact that any pseudo-Boolean (Heyting) algebra for intuitionistic logic can be used to construct a dynamic model which is equivalent to the pseudo-Boolean algebra in the sense that an element is identical with the unit element in the pseudo-Boolean algebra if and only if the corresponding proposition is a tautology in the dynamic model (this construction is presented in Gärdenfors 1985). Cf. van Benthem (1992) for further connections between various kinds of algebras and dynamic models.

[11] This section is, to a large extent, borrowed from Balkenius and Gärdenfors (1991).

paper, however, I will not consider learning processes in neural networks.

3.1 Schemata and resonant states in neural networks

First of all we need a general description of neural networks. One can define a neural network N as a 4-tuple <S,F,C,G>. Here S is the space of all possible *states* of the neural network. The dimensionality of S corresponds to the number of parameters used to describe a state of the system. Usually $S=[a,b]^n$, where [a,b] is the working range of each neuron and n is the number of neurons in the system. We will assume that each neuron can take excitatory levels between 0 and 1. This means that a state in S can be described as a vector $x = <x_1,...,x_n>$ where $0 \leq x_i \leq 1$, for all $1 \leq i \leq n$. The network N is said to be binary if $x_i = 0$ or $x_i = 1$ for all i, that is if each neuron can only be in two excitatory levels.

C is the set of possible *configurations* of the network. A configuration c □ C describes for each pair i and j of neurons the connection c_{ij} between i and j. The value of c_{ij} can be positive or negative. When it is positive the connection is *excitatory* and when it is negative it is *inhibitory*. A configuration c is said to be *symmetric* if $c_{ij} = c_{ji}$ for all i and j.

F is a set of *state transition functions* or *activation functions*. For a given configuration c □ C, a function f_c □ F describes how the neuron activities spread through that network. G is a set of *learning functions* which describe how the configurations develop as a result of various inputs to the network.

By changing the behavior of the functions in the two sets F and G, it is possible to describe a large set of different neural mechanisms. In the rest of the section, I will assume that the state in C is fixed while studying the state transitions in S. This means that I will not consider the effects of learning within a neural network.

In Balkenius (1990) and Balkenius and Gärdenfors (1991) it is argued that there is a very simple way of defining the notion of a *schema* within the theory of neural networks that can be seen as a generalization of the notion of a proposition. The definition proposed there is that a schema α corresponds to a vector $<\alpha_1,...,\alpha_n>$ in the state space S. That a schema α is currently *represented* (or *accepted*) in a neural network with an activity vector $x = <x_1,...,x_n>$ means that $x_i \geq \alpha_i$, for all $1 \leq i \leq n$. There is a natural way of defining a partial order of 'greater informational content' among schemata by putting $\alpha \geq \beta$ iff $\alpha_i \geq \beta_i$ for all $1 \leq i \leq n$. There is a minimal schema in this ordering, namely $0 = <0,...,0>$ and a maximal element $1 = <1,...,1>$.

In the light of this definition, let us consider some general desiderata for schemata. Firstly, it is clear that depending on what the activity patterns in a neural network correspond to, schemata as defined here can be used for representing objects, situations, and actions. [12]

Secondly, if $\alpha \geq \beta$, then β can be considered to be a more *general* schema than α and α can thus be seen as an *instantiation* of the schema β. The part of α not in β, is a *variable* instantiation of the schema β. This implies that all schemata with more information than β can be considered to be an instantiation of β with different variable instantiations. Thus, schemata can have variables even though they do not have any *explicit* representation of variables.[13] Only the *value* of the variable is represented and not the variable as such. In general, it can be said that the view on schemata presented here replaces symbols by vectors representing various forms of *patterns*.

Thirdly, it will soon be shown that schemata support *default assumptions* about the environment. The neural network is thus capable of filling in missing information.

The abstract definition of schemata presented here fits well with Smolensky's (1986) analysis of schemata in terms of 'peaks' in a harmony function. And in Smolensky (1991a, p. 202) he says that his treatment of connectionism

> "is committed to the hypothesis that mental representations are *vectors* partially specifying the state of a dynamical system (the activities of units in a connectionist network), and that mental processes are specified by the dynamical equations governing the evolution of that dynamical system."

Some interesting examples of schemata are found in Rumelhart, Smolensky, McClelland and Hinton (1986) who address, among other things, the case of schemata for rooms. The network they investigate contains 40 neurons representing microfeatures of rooms like has-ceiling, contains-table, etc.[14] There are no neurons in the network representing kitchens and bedrooms, but various rooms can be represented implicitly as schemata of the network; the peaks of the harmony function correspond to prototypical rooms.

At this point, I want to emphasize that the definition of schemata given here is the simplest possible and is introduced just to show that even with elementary means it is possible to exhibit the compositional and

[12] For some examples of this, cf. Balkenius 1992.

[13] Smolensky's (1991b) solution to the problem of variables is more complicated and to some extent *ad hoc*. On the other hand, he can handle asymmetric relations and some embedding features that cannot be given a simple analysis on the present approach.

[14] The network is presented on pp. 22–24 in Rumelhart, Smolensky, McClelland and Hinton (1986).

systematic structure desired by the adherents of the symbolic paradigm. The definition applies to any neural network falling under the general description above. However, for networks that are designed for some special purpose, it is possible to introduce more sophisticated and fine-structured definitions of schemata that better capture what the network is intended to represent.

There are some elementary operations on schemata as defined above that will be of interest when I consider nonmonotonic inferences in a neural network. The first operator is the *conjunction* $\alpha \bullet \beta$ of two schemata $\alpha = <\alpha_1,...,\alpha_n>$ and $\beta = <\beta_1,...,\beta_n>$ which is defined as $<\gamma_1,...,\gamma_n>$, where $\gamma_i = \max(\alpha_i,\beta_i)$ for all i. If schemata are considered as corresponding to observations in an environment, one can interpret $\alpha \bullet \beta$ as the *coincidence* of two schemata, i.e., the simultaneous observation of two schemata.

Secondly, the *complement* α^* of a schema $\alpha = <\alpha_1,...\alpha_n>$ is defined as $<1–\alpha_1,...,1–\alpha_n>$ (recall that 1 is assumed to be the maximum activation level of the neurons, and 0 the minimum). In general, the complementation operation does not behave like negation since, for example, if $\alpha = <0.5,...,0.5>$, then $\alpha^* = \alpha$. However, if the neural network is assumed to be binary, that is if neurons only take activity values 1 or 0, then * will indeed behave as a classical negation on the class of binary-valued schemas.

Furthermore, the interpretation of the complement is different from the classical negation since the activities of the neurons only represent *positive* information about certain features of the environment. The complement α^* reflects a lack of positive information about α. It can be interpreted as a schema corresponding to the observation of everything but α. As a consequence of this distinction it is pointless to define implication from conjunction and complement. The intuitive reason is that it is impossible to observe an implication directly. A consequence is that the ordering $\geq$ only reflects greater *positive* informational content.

Finally, the *disjunction* $\alpha H \beta$ of two schemata $\alpha = <\alpha_1,...,\alpha_n>$ and $\beta = <\beta_1,...,\beta_n>$ is defined as $<\gamma_1,...,\gamma_n>$, where $\gamma_i = \min(\alpha_i,\beta_i)$ for all i. The term 'disjunction' is appropriate for this operation only if we consider schemata as representing propositional information. Another interpretation that is more congenial to the standard way of looking at neural networks is to see α and β as two instances of a *variable.* $\alpha H \beta$ can then be interpreted as the *generalization* from these two instances to an underlying variable.

It is trivial to verify that the De Morgan laws $\alpha H \beta = (\alpha^* \bullet \beta^*)^*$ and $\alpha \bullet \beta = (\alpha^* H \beta^*)^*$ hold for these operations. The set of all schemata forms a distributive lattice with zero and unit, as is easily shown. It is a boolean algebra, whenever the underlying neural network is binary. In this way we have already identified something that can be viewed as a *syntactic* (and *compositional*) structure on the set of *vectors* representing schemata.

How does the structure on states of networks, generated by the operators •, H, and * relate to the postulates (*P1*) – (*P7*) in Section 2? For each schema α, it is trivial to define a function on activity states $x = <x_1,...,x_n>$ of a network, corresponding to giving α as an input, by putting $\alpha(x) = \alpha \bullet x$, for all x in S. Then if we put 3 = •, £ = H, Å = 0 and ⊥ = 1 it is easy to verify that postulates (*P1*) – (*P4*) and (*P6*) – (*P7*) hold. On the other hand, there does not seem to be any operator on vectors that is an equalizer. The candidate from classical logic, i.e., $(\alpha \bullet \beta)H(\alpha^* \bullet \beta^*)$, does not satisfy the requirements of (*P5*) in general. However, in the special case when the network is binary, this construction works and all of the postulates (*P1*) – (*P8*) are satisfied.

3.2 Nonmonotonic inferences in a neural network

A desirable property of a network that can be interpreted as performing *inferences* of some kind is that it, when given a certain input, stabilizes in a state containing the results of the inference. In the theory of neural networks such states are called *resonant states*. In order to give a precise definition of this notion, consider a neural network N = <S,F,C,G>. Let us assume that the configuration c is fixed so that we only have to consider one state transition function $f = f_c$. Let $f^0(x) = f(x)$ and $f^{n+1}(x) = f\square f^n(x)$. Then a state y in S is called *resonant* if it has the following properties:

(i) $f(y) = y$ (equilibrium)

(ii) If for any x [S and each $\varepsilon > 0$ there exists a $\delta > 0$ such that $|x–y| < \delta$, then $|f^n(x)–y| < \varepsilon$ when $n \geq 0$ (stability)

(iii) There exists a δ such that if $|x–y| < \delta$, then $\lim_{n\varnothing\infty} f^n(x) = y$ (asymptotic stability).

Here |.| denotes the standard euclidean metric on the state space S. A neural system N is called *resonant* if for each x in S there exists a $n > 0$, that depends only on x, such that $f^n(x)$ is a resonant state.

If $\lim_{n\varnothing\infty} f^n(x)$ exists, it is denoted by [x] and [.] is called the *resonance function* for the configuration c. It follows from the definitions above that all resonant systems have a resonance function. For a resonant system we can then define *resonance equivalence* as x~y iff [x]=[y]. It follows that ~ is an equivalence relation on S that partitions S into a set of equivalence classes.

It can be shown (Cohen and Grossberg 1983, Grossberg 1989) that a large class of neural networks have resonance functions. A common feature of these types of neural networks is that they are based on *symmetrical* configuration functions c, that is, the connections between two neurons are equal in both directions.

The function [.] can be interpreted as filling in *default* assumptions about the environment, so that the schema represented by [α] contains information about what the network *expects* to hold when given α as input. Even if α only gives a partial description of, for example, an object, the neural network is capable of supplying the missing information in attaining the resonant state [α]. The expectations are determined by the configuration function c, and thus expectations are 'equilibirum' features of a network in contrast to the transient input α.

I now turn to the problem of providing a different perspective of the activities of a neural network which will show it to perform *nonmonotonic inferences*. A first idea for describing the nonmonotonic inferences performed by a neural network N is to say that the resonant state [α] represents the expectations of the network given the input information represented by α. The expectations can also be described as the set of nonmonotonic conclusions to be drawn from α. However, the schema α is not always included in [α], that is, $[\alpha] \geq \alpha$ does not hold in general. Sometimes a neural network *rejects* parts of the input information – in pictorial terms it does not always believe what it sees.

So if we want α to be included in the resulting resonant state, one has to modify the definition. The most natural solution is to 'clamp' α in the network, that is to add the *constraint* that the activity levels of all neurons is above α_i, for all i. Formally, we obtain this by first defining a function f_α via the equation $f_\alpha(x) = f(x)\bullet\alpha$ for all x [S. We can then, for any resonant system, introduce the function $[.]^\alpha$ (for a fixed configuration c [C) as follows:

$$[x]^\alpha = \lim_{n\varnothing\infty} f_\alpha{}^n(x)$$

This function will result in resonant states for the same neural networks as for the function [.].

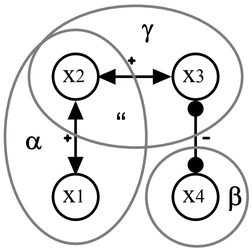
The key idea of this section is then to define a nonmonotonic inference relation □ between schemata in the following way:

α □ β iff $[\alpha]^\alpha \geq \beta$

This definition fits very well with the interpretation that nonmonotonic inferences are based on the dynamics of information as developed in Section 2. Note that α and β in the definition of □ have double interpretations: From the *connectionist* perspective, they are schemata which are defined in terms of activity vectors in a neural network. From the other perspective, the *symbolic*, they are viewed as formal expressions with a grammatical structure. Thus, in the terminology of Smolensky (1988), we make the transition from the subsymbolic level to the symbolic simply by giving a different *interpretation* of the structure of a neural network. Unlike Fodor and Pylyshyn (1988), we need not assume two different systems handling the two different levels. In contrast, the symbolic level *emerges* from the subsymbolic in one and the same system.

Before turning to an investigation of the general properties of □ generated by the definition, I will illustrate it by showing how it operates for a simple neural network.

Example: The network depicted below consists of four neurons with activities $x_1,...,x_4$. Neurons that interact are connected by lines. Arrows at the ends of the lines indicate that the neurons excite each other; dots indicate that they inhibit each other. If we consider only schemata corresponding to binary activity vectors, it is possible to identify schemata with *sets* of active neurons. Let three schemata α, β, and γ correspond to the following activity vectors: α=<1 1 0 0>, β=<0 0 0 1>, γ=<0 1 1 0>. Assume that x_4 inhibits x_3 more than x_2 excites x_3. If α is given as input, the network will activate x_3 and thus γ. It follows that $[\alpha]^\alpha \geq \gamma$ and hence α □ γ. Extending the input to $\alpha\bullet\beta$ causes the network to withdraw γ, i.e., no longer represent this schema, since the activity in x_4 inhibits x_3. In formal terms $\alpha\bullet\beta$ ¸ γ.



One way of characterizing the nonmonotonic inferences generated by a neural network is to study them in terms of general *postulates* for nonmonotonic logics that have been investigated in the literature (Gabbay 1985, Makinson 1993, Kraus, Lehmann and Magidor 1991, Makinson and Gärdenfors 1991, Gärdenfors and Makinson 1993).

It follows immediately from the definition of $[.]^\alpha$ that □ satisfies the property of *Reflexivity*:

a □ a

If we say that a schema β follows logically from α, in symbols α 7 β, just when $\alpha \geq \beta$, then it is also trivial to verify that □ satisfies *Supraclassicality*:

If α 7 β, then α □ β

In words, this property means that immediate consequences of a schema are also nonmonotonic consequences of the schema.

If we turn to the operations on schemata, the following postulate for conjunction is also trivial:

If α □ β and α □ γ, then α □ β•γ (*And*)

More interesting are the following two properties:

If α □ β and α•β □ γ, then α □ γ (*Cut*)

If α □ β and α □ γ, then α•β □ γ (*Cautious Monotony*)

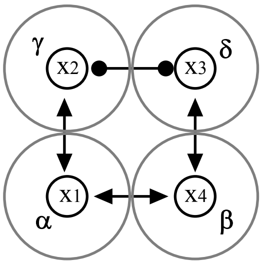
Together Cut and Cautious Monotony are equivalent to each of the following postulates:

If α □ β and β 7 α, then α □ γ iff β □ γ (*Cumulativity*)

If α □ β and β □ α, then α □ γ iff β □ γ (*Reciprocity*)

Cumulativity has become an important touchstone for nonmonotonic systems (Gabbay 1985, Makinson 1993). It is therefore interesting to see that the inference operation defined here seems to satisfy Cumulativity (and thus Reciprocity) for almost all neural networks where it is defined. However, it is possible to find some cases where it is not satisfied:

Counterexample to Reciprocity: The network illustrated below is a simple example of a network that does not satisfy Reciprocity (or Cumulativity). For this network it is assumed that all inputs to a neuron are simply added. If we assume that there is a strong excitatory connection between α and β, it follows that α □ β and β □ α since α and β do not receive any inhibitory inputs. Suppose that α = <1 0 0 0> is given as input. From the assumption that the inputs to x_3 interact *additively* it follows that γ receives a larger input than δ, because of the time delay before δ gets activated. If the inhibitory connection between γ and δ is large, the excitatory input from β can never effect the activity of x_3. We then have α □ γ and α ¸ δ. If instead β = <0 1 0 0> is given as input, the situation is the opposite, and so δ gets excited but not γ, and consequently α ¸ γ and α □ δ. Thus, the network does not satisfy Reciprocity.



A critical factor here seems to be the *linear* summation of inputs that ’locks’ x_2 and x_3 to inputs from the outside because the inhibitory connection between them is large.

Extensive computer simulations have been performed with networks that obey ’shunting’ rather than linear summation of excitatory and inhibitory inputs (see Balkenius and Gärdenfors 1991). They suggest that reciprocity is satisfied in all networks that obey this kind of interaction of the inputs.

Shunting interaction of inputs is used in many biologically inspired neural network models (e.g., Cohen and Grossberg 1983, Grossberg 1989) and is an approximation of the membrane equations of neurons. A simple example of such a network can be described by the following equation:

$$x_i(t+1) = x_i(t) + \delta(1-x_i(t))\Sigma_j d(x_i(t))c_{ji}^{+} + \delta x_i(t)\Sigma_j d(x_i(t))c_{ji}^{-}$$

Here δ is a small constant, c_{ij}^{+} and c_{ij}^{-} are matrices with all $c_{ij}^{+}=c_{ji}^{+}\geq 0$, and all $c_{ij}^{-}=c_{ji}^{-}\leq 0$; $d(x)\geq 0$ and $d'(x)>0$. The positive inputs to neuron x_i are shunted by the term $(1-x_i(t))$ and the negative inputs by $x_i(t)$. As a consequence, the situation where one input locks another of opposite sign cannot occur, in contrast to the linear case above. In other words, a change of input, that is a change in $\Sigma_j d(x_i(t))c_{ji}^{+}$ or $\Sigma_j d(x_i(t))c_{ji}^{-}$, will always change the equilibrium of x_i. The fact that one input never locks another of opposite sign seems to be the reason why all the simulated shunting networks satisfy Reciprocity.

For the disjunction operation it does not seem possible to show that any genuinely new postulates are fulfilled. The following special form of transitivity is a consequence of Cumulativity (cf. Kraus, Lehmann, and Magidor 1991, p. 179):

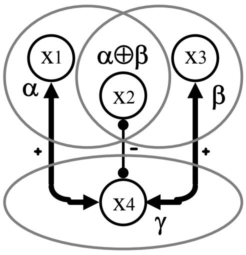
If αHβ □ α and α □ γ, then αHβ □ γ

This principle is thus satisfied whenever Cumulativity is.

The general form of Transitivity, i.e., if α □ β and β □ γ, then α □ γ, is not valid for all α, β, and γ, as can be shown by the first example above. Nor is *Or* generally valid:

If α □ γ and β □ γ, then αHβ □ γ *(Or)*

Counterexample to Or: The following network is a simple counterexample: x_1 excites x_4 more than x_2 inhibits x_4. The same is true for x_3 and x_2. Giving $\alpha = <1\ 1\ 0\ 0>$ or $\beta = <0\ 1\ 1\ 0>$ as input activates x_4, thus α □ γ and β □ γ. On the other hand, the neuron x_2 which represents schema $\alpha H \beta$ has only inhibitory connections to x_4. As a consequence $\alpha H \beta$ ¸ γ.



In summary, following Balkenius and Gärdenfors (1991), it has been shown that by introducing an appropriate schema concept and exploiting the higher-level features of a resonance function in a neural network it is possible to define a form of nonmonotonic inference relation. It has also been established that this inference relation satisfies some of the most fundamental postulates for nonmonotonic logics.

The construction presented in this section is thus an example of how symbolic features can emerge from the subsymbolic level of a neural network. However, the notion of inference presented here is clearly part of the associationist tradition, since it is based on other primitive notions than what is common within the symbolic paradigm. Sellars (1980, p. 265) discusses six conceptions of the status of material rules of inference. The notion presented here fits well with his sixth:

> Trains of thought which are said to be governed by “material rules of inference” are actually *not inferences at all*, but rather activated associations which mimic inference, concealing their intellectual nudity with stolen “therefores”.

4. Symbolic and subsymbolic: two perspectives on the same process

With the two examples from Sections 2 and 3 in mind, I now want to turn to the current discussion within cognitive science concerning the relation between symbolic and subsymbolic processes. It should be clear that what is normally considered to be a subsymbolic process fits well with the associationist paradigm outlined in Section 1. The main thesis of this article is that the symbolic and subsymbolic approaches are not two rival paradigms of computing, rather they are best viewed as two different perspectives that can be adopted when describing the activities of various computational devices.[15] Smolensky (1991a) formulates the same idea as follows:

> ”Rather than having to model the mind as *either* a /symbolic/ structure cruncher *or* a number cruncher, we can now see it as a number cruncher in which the numbers crunched are in fact representing complex /symbolic/ structures.” (pp. 215–216)

> ”The connectionist cognitive architecture is intrinsically two-level: Semantic interpretation is carried out at the level of patterns of activity while the complete, precise, and formal account of mental processing must be carried out at the level of individual activity values and connections. Mental processes reside at a lower level of analysis than mental representations.” (p. 223)

In the light of this thesis let us look at some of the major discussions concerning the relation between symbolic processing and connectionism, namely, Smolensky (1988, 1991a), Fodor and Pylyshyn (1988) and Fodor and McLaughlin (1990).

4.1 What is the proper treatment of connectionism?

Smolensky’s (1988) ’proper treatment of connectionism’ (PTC) seems quite closely related to my position. When he argues that the symbolic and the subsymbolic paradigms are incompatible, as I outlined in Section 1.3, my interpretation is that he says that the two perspectives cannot be adopted to one level only. As we shall see below, this seems to be exactly what Fodor and Pylyshyn (1988) try to do.

There are several aspects of Smolensky’s analysis that are similar to the one presented here. For instance, at the end of the article he describes his position as ’emergentist’.[16] And, as mentioned in Section 3.1, his

[15] In Gärdenfors (1992) I argue that in order to understand inductive reasoning, and thereby cognition in general, one must distinguish between at least three levels: the symbolic (there called linguistic), the conceptual, and the subconceptual level. The distinction between the conceptual and the subconceptual levels is not important for the purposes of the present paper; they can both be seen as subsymbolic. (However, as pointed out in Gärdenfors (1992), Smolensky (1988) confuses the symbolic and the conceptual levels).

[16] Also cf. Woodfield and Morton's (1988) commentary and the Author's Response on p. 64. In Smolensky (1991a, p. 202) he writes: “In giving up symbolic computation to undertake connectionist modeling, we connectionists have taken out an enormous loan, on which we are still paying nearly all interest: solving the basic problems we have created for ourselves rather than solving the problems of cognition. In my view the loan is worth taking out for the

analysis of schemata is congenial with the one presented there. Furthermore, at a first glance at least, his description of inference on the subsymbolic level seems to be quite similar to the account presented in Section 3.2:

> ”A natural way to look at the knowledge stored in connections is to view each connection as a *soft constraint*. ... Formalizing knowledge in soft constraints rather than hard rules has important consequences. Hard constraints have consequences singly; they are rules that can be applied separately and sequentially – the operation of each proceeding independently of whatever other rules may exist. But soft constraints have no implications singly; any one can be overridden by the others. It is only the entire set of soft constraints that has any implications. Inference must be a cooperative process, like the parallel relaxation processes typically found in subsymbolic systems. Furthermore, adding additional soft constraints can repeal conclusions that were formerly valid: Subsymbolic inference is fundamentally nonmonotonic” (1988, p. 18).

However, one worry I have with this description of the activities of a connectionist system is that Smolensky still sees it as performing *inferences* even on the subconceptual level.[17] This point is made very clearly in Dellarosa’s (1988, p. 29) commentary:

> ”It is a belief of many cognitive scientists (most notably, Fodor 1975) that the fundamental process of cognition is inference, a process to which symbolic modelling is particularly well suited. While Smolensky points out that statistical inference replaces logical inference in connectionist systems, he too continues to place inference at the heart of all cognitive activity. I believe that something more fundamental is taking place. In most connectionist models, the fundamental process of cognition is not inference, but is instead the (dear to the heart of psychologists) activation of associated units in a network. Inference ’emerges’ as a system-level interpretation of this microlevel activity, but – when representations are distributed – no simple one-to-one mapping of activity patterns to symbols and inferences can be made. From this viewpoint, the fundamental process of cognition is the activation of associated units, and inference is a second-order process.”

Thus Smolensky is wrong in talking about ’nonmonotonic inferences’ on the subsymbolic level, since there are no inferences on this level; claiming this is basically a kind of category error. However, as has been argued in the previous section, he is right in that the inferences that emerge from the subsymbolic processes on the symbolic level are fundamentally nonmonotonic.

It should be noted that two perspectives on computing that are discussed here are not only applicable to neural networks. Also the behavior of a traditional computer with a von Neumann architecture can be given a ’subsymbolic’ interpretation and need not be seen as merely symbol crunching. The subsymbolic perspective is adopted when one describes the general properties of the physical processes driving the computer; for example when describing the electric properties of transistors. This is the perspective that one must adopt when the computer is defective, in which case the processing on the symbolic level does not function as expected.[18]

A consequence of the fact that one can adopt two perspectives on all kinds of computing devices is that every ascription of symbolic processing to some system is an *interpretation* of the subsymbolic activities. The Turing paradigm of computation neglects this distinction since a computer is thought to uniquely identify some Turing machine; and Turing machines are clearly described on the symbolic level.[19] The reason this identification works is that traditional computers are constructed to be ’digital’, i.e., on the subsymbolic perspective the outcomes of the electronic processes are very robust with respect to disturbances so that particular currents can be identified as either ’1’s or ’0’s. However, the identification may break down as soon as the computer is malfunctioning.

It follows that the notion of ’computation’ can be given two meanings. The first, and to many the only meaning is computation on the symbolic level in the

goal of understanding how symbolic computation, or approximations of it, can emerge from numerical computation in a class of dynamical systems sharing the most general characteristics of neural computation.”

[17] Fodor and Pylyshyn (1988, pp. 29–30) too, make inferences the engine of cognition: “It would not be unreasonable to describe Classical Cognitive Science as an extended attempt to apply the methods of proof theory to the modeling of thought (and similarly, of whatever mental processes are plausibly viewed as involving inferences; preeminently learning and perception).”

[18] In this context, the subsymbolic perspective is related to adopting the ’physical stance’ in the terminology of Dennett (1978), while the symbolic level then corresponds to the ’design stance’. The analogy is not perfect since the subsymbolic perspective on the function of a computer need not be tied to a particular physical realization, but can be kept at the level of general functional properties of e.g., transistors, independently of what material they are made from. The same argument, of course, applies to neural networks, the subsymbolic level of which can be described independently of their physical level. The upshot is that the subsymbolic perspective falls ’between’ the design stance and the physical stance.

[19] Cf. the quotation from Fodor (1981) in Section 1.1.

sense that is made precise by ’Turing computable’. According to Church’s thesis this kind of computation is all there is *on the symbolic level*. The other sense of computation only becomes apparent when one adopts a subsymbolic (connectionist or more general associationist) perspective. From this perspective ’computation’ means ’processing representations’, where the representations have a fundamentally different structure compared to those on the symbolic level. And processing on this level does not mean ’manipulating symbols’, but must be characterized in other ways. Some kinds of processing of representations on the subsymbolic level generate structures that can be interpreted meaningfully on the symbolic level. However, there are also many kinds of processes that cannot be interpreted on the symbolic level as performing any form of Turing computation. For instance, the notion of ’analog’ computation only makes sense on the subsymbolic level. Hence, the class of computational processes on the subsymbolic level is much wider than the class of processes corresponding to Turing computations. Thus, Church’s thesis does not apply to this sense of ’computation’.

4.2 The compatibility of symbolism and connectionism

Let me finally return to Fodor and Pylyshyn’s (1988) argument against the systematicity and compositionality of connectionism. Their main conclusion is that since cognition is compositional and systematic and since connectionist systems lack those properties, while ’Classical’, i.e., symbolic, systems have them, it is only symbolic systems that can represent cognitive processes.

First of all, it should be noted that they assume that, even if there are several levels of analysis, all ’cognitive’ levels are representational:

> ”Since Classicists and Connectionists are both Representationalists, for them any level at which states of the system are taken to encode properties of the world counts as a *cognitive* level; and no other levels do.” (Fodor and Pylyshyn 1988, p. 9)

According to them, this assumption about a unique representational level puts a strait-jacket on connectionist methodology:

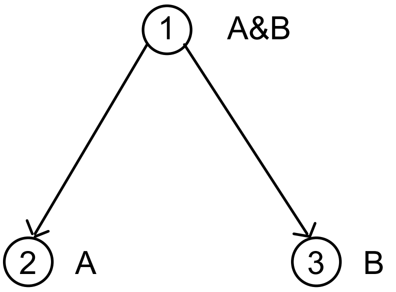
> ”It is, for example, *no use at all*, from the cognitive psychologist’s point of view, to show that the *non*representational (e.g. neurological, or molecular, or quantum mechanical) states of an organism constitute a Connectionist network, because that would *leave open* the the question whether the mind is such a network *at the psychological level*. It is, in particular, perfectly possible that nonrepresentational neurological states are interconnected in the ways described by Connectionist models *but that the representational states themselves are not*” (p. 10).

So, the key question becomes: How do connectionist systems represent? Fodor and Pylyshyn summarize the disparity between symbolic (’Classical’) and connectionist systems as follows:

> ”Classical and Connectionist theories disagree about the nature of mental representation; for the former, but not for the latter, mental representations characteristically exhibit a combinatorial constituent structure and a combinatorial semantics. Classical and Connectionist theories also disagree about the nature of mental processes; for the former, but not for the latter, mental processes are characteristically sensitive to the combinatorial structure of the representations on which they operate” (1988, p. 32).

Their main argument for why connectionist systems exhibit neither compositionality (i.e., combinatorial constituent structure) nor systematicity (i.e., sensitivity to this combinatorial structure in processes) is based on their interpretation of how networks represent. It is at this point that they seem to be confusing the symbolic and the connectionist (associationist) perspectives. On p. 12 they state that “[r]oughly, Connectionists assign semantic content to ’nodes’ [neurons] ... – i.e., to the sorts of things that are typically labeled in Connectionist diagrams; whereas Classicists assign semantic content to *expressions* – i.e., to the sort of things that get written on the tapes of Turing machines and stored at addresses in von Neumann machines.”

Fodor and Pylyshyn’s paradigm example of such an assignment is presented on pp. 15–16, where they consider the difference between a connectionist machine handling an inference from *A&B* to *A* and a symbolic machine doing the same thing. They assume that the connectionist machine consists of a network of “labelled nodes” that looks as follows (their Figure 2):



In this network "[d]rawing an inference from *A&B* to *A* thus corresponds to an excitation of node 2 being caused by an excitation of node 1" (p. 15).[20] The fundamental mistake in this example is that disjoint nodes are assumed to *represent* different expressions.[21] This assumption conflates representations on the symbolic level (which is where representations of *expressions* and *inferences* belong) with representations on the connectionist level (where representations of associations are handled). Smolensky (1991a, p. 206) argues concerning Fodor and Pylyshyn's example that

> ”it is a serious mistake to view this as the paradigmatic connectionist account for anything like human inferences of this sort. The kind of *ultralocal* connectionist representation, in which entire propositions are represented by individual nodes, is far from typical of connectionist models, and certainly not to be taken as *definitive* of the connectionist approach."

Given this assumption, it is no wonder that Fodor and Pylyshyn can then argue that networks don't exhibit compositionality and systematicity.

The best way to rebut their argument is to provide a constructive counterexample, i.e., an example of a connectionist system representing things with a compositional and systematic structure. This is readily available from the account of representation and inference in neural networks presented in Section 3. There it is *schemata* that represent. Schemata pick out certain *patterns of activities* in the nodes of a connectionist system. As is shown in Section 3.2, it is trivial to define some elementary operations on schemata. These operations immediately endow a *compositional* structure on schemata and the components of schemata can be related and combined in a *systematic* way (unlike the one-node representations in Fodor and Pylyshyn's examples).[22] For example, if the schemata $\alpha \bullet \beta$ and $\gamma H \delta$ are both represented in a particular state of a network, one can meaningfully ask whether schemata like $\alpha \bullet \delta$ or $\gamma H \beta$ are also represented in that state (the latter always is). Admittedly, the compositional structure is not very spectacular from a cognitive point of view, but what more can be expected from such a simplistic construction?[23] Furthermore, what is at stake here is not the richness of the representations, but the mere possibility of endowing connectionist systems with a compositional structure of representations.

To be sure, the schemata do not have an *explicit* symbolic structure in the sense that somewhere in the network one finds expressions referring to the schemata (or something representing such expressions). Fodor and Pylyshyn (1988, p. 33) seem to think that any productive representational system, i.e., a finitely generated system capable of representing an infinite number of object, must be a symbol system (cf. Bernsen and Ulbæk 1992a,b). However, according to the definition of schemata given in Section 3.2, a network with a finite number of nodes can *implicitly* represent an infinite number of schemata.[24] And that is sufficient to establish the productivity of this kind of representation. A similar point is made by Smolensky in his (1991a).

[20] A closely related example is given on pp. 47–48 in their paper.

[21] They provide a similar argument on p. 49: "What is deeply wrong with Connectionist architecture is this: Because it acknowledges neither syntactic nor semantic structure in mental representations, it perforce treats them not as a generated set but as a list." However, in fairness it should be acknowledged that Fodor and Pylyshyn (1988, p. 12 footnote 7) consider the possibility of having aggregates of neurons representing expressions: "But a subtler reading of Connectionist machines might take it to be the total machine *states* that have content, e.g. the state of *having such and such a node excited*." Even though this comes close to the schema representation presented in Section 3.1, they also claim that "[m]ost of the time the distinction between these two ways of talking does not matter for our purposes" (*ibid*.). It certainly does, as shall be argued shortly.

[22] Cf. Smolensky (1991a, p. 211): "Thus in the distributed case, the relation between the node of /the figure above/ labeled A&B and the others is one kind of whole/part relation. An inference mechanism that takes as input the vector representing A&B and produces as output the vector representing A is a mechanism that extracts a part from a whole. And in this sense it is no different from a symbolic inference mechanism that takes the syntactic structure A & B and extracts from it the syntactic constituent A. The connectionist mechanisms for doing this are of course quite different than the symbolic mechanisms, and the approximate nature of the whole/part relation gives the connectionist computation different overall characteristics: we don't have simply a new implementation of the old computation."

[23] Another type of example is provided by Bernsen and Ulbæk (1992a,b), who deal with systematicity in representations of spatial relations.

[24] If the neurons can only take a finite number of activity levels, the *references* of the schemata, described as vectors of activities, will only constitute a finite class. However, the same applies, for example, to classical propositional logic generated from a finite number of atomic sentences: Even if the language contains an infinite number of formulas, their references, i.e., the propositions expressed (i.e., truth-functions), are finite in number.

Fodor and McLaughlin (1990) have challenged the proposal that vectorial representations in connectionist systems can exhibit systematicity and productivity. The gist of their argument seems to be the following (p. 200):

> ”… the components of tensor product and superposition vectors differ from Classical constituents in the following way: when a complex Classical symbol is tokened, its constituents are tokened. When a tensor product vector or superposition vector is tokened, its components are not (except per accidens). The implication of this difference, from the point of view of the theory of mental processes, is that whereas the Classical constituents of a complex symbol are, ipso facto, available to contribute to the causal consequences of its tokenings – in particular, they are available to provide domains for mental processes – the components of tensor product and superposition vectors can have no causal status as such. What is merely imaginary can’t make things happen, to put this point in a nutshell.”

However, the notion of causality Fodor and McLaughlin presumes here is very odd, to say the least – they assume that ’tokenings’ of symbols completely decide the causal structure of the mental processes. The subsymbolic processes can, according to them, have no causal role since they are not tokened. To me this seems like saying that the tokenings of ’123’, ’∞’, ’45’, and ’=’ on a pocket calculator are the only causes of a tokening of ’5535’ appearing in the window, while the underlying electronic processes, not being tokened, can play no causal role.

On the contrary, if we want to analyse the causality of mental processes, we should focus on the subsymbolic level or even the underlying physical processes, while the emerging symbolic structures will, in themselves, not be causally efficacious.[25] Consequently, I believe that Fodor and McLaughlin’s attempt to save the argument that the connectionist approach is not a viable explanation of mental processes is a dead end.

In summary, Fodor and Pylyshyn (and McLaughlin) have put blinders on themselves by only considering a special type of representations in connectionist systems. Given the ensuing narrow field of vision, they can argue that connectionist systems cannot represent what is required for modelling cognition. However, I have argued that once one is allowed to view a wider class of representational possibilities, like, e.g., the schemata of Section 3.2, the limitations they point out are no longer there (this is not to say that there are no limitations).

ACKNOWLEDGEMENTS

I wish to thank Christian Balkenius, Johan van Benthem, Linus Broström, Jan van Eijck, Kenneth Holmqvist, and the Cognitive Science Group in Lund for helpful discussions. My work with this article has been supported by the Swedish Council for Research in the Humanities and Social Sciences.

REFERENCES:

Balkenius, C. (1992): “Neural mechanisms for self-organization of emergent schemata, dynamical schema processing, and semantic constraint satisfaction,” manuscript, *Lund University Cognitive Studies* No. 14, Lund University.

Balkenius, C. and P. Gärdenfors (1991): “Nonmonotonic inferences in neural networks,” pp. 32–39 in *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, J.A. Allen, R. Fikes, and E. Sandewall, eds. (San Mateo, CA: Morgan Kaufmann).

Beale, R. and T. Jackson (1990): *Neural Computing: An Introduction*, Bristol: Adam Hilger.

Benthem, J. van (1992): “Logic and the flow of information,” to appear in the *Proceedings of the 9th International Congress of Logic, Methodology, and Philosophy of Science*, Amsterdam: North-Holland.

Bernsen, N. O. and I. Ulbæk (1992a): “Two games in Town: Systematicity in distributed connectionist systems”, *AISBQ Special Issue on Hybrid Models of Cognition* Part 2, No. 79, 25–30.

Bernsen, N. O. and I. Ulbæk (1992b): “Systematicity, thought and attention in a distributed connectionist system,” manuscript, Centre of Cognitive Science, Roskilde University.

[25] Smolensky (1991a, pp. 222–23) makes this point in the following way: “The Classical strategy for explaining the systematicity of thought is to hypothesize that there is a precise formal account of the cognitive architecture in which the constituents of mental representations have causally efficacious roles in the mental processes acting on them. The PTC view denies that such an account of the cognitive architecture exists, and hypothesizes instead that, like the constituents of structures in quantum mechanics, the systematic effects observed in the processing of mental representations arises because the evolution of vectors can be (at least partially and approximately) explained in terms of the evolution of their components, even though the precise dynamical equations apply at the lower level of the individual numbers comprising the vectors and cannot be pulled up to provide a precise temporal account of the processing at the level of entire constituents – i.e., even though the constituents are not causally efficacious.”

Churchland, P. S. (1986): Neurophilosophy: Toward a Unified Science of the Mind/Brain, Cambridge, MA: MIT Press.

Cohen, M. A. and S. Grossberg (1983): "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks," *IEEE Transactions on Systems, Man, and Cybernetics*, SMC–13, 815–826.

Dellarosa, D. (1988): "The psychological appeal of connectionism," *Behavioral and Brain Sciences 11:1*, 28–29.

Dennett, D. (1978): *Brainstorms*, Cambridge, MA: MIT Press.

Eijck, J. van, and F. - J. de Vries (1992): "Dynamic interpretation and Hoare deduction," *Journal of Logic, Language and Information 1*, 1–44.

Fodor, J. (1975): *The Language of Thought*, Cambridge, MA: Harvard University Press.

Fodor, J. (1981): *Representations*, Cambridge, MA: MIT Press.

Fodor, J. and B. P. McLaughlin (1990): "Connectionism and the problem of systematicity: Why Smolensky's solution doesn't work," *Cognition 35*, 183–204.

Fodor, J. and Z. Pylyshyn (1988): "Connectionism and cognitive architecture: A critical analysis," in. S. Pinker & J. Mehler (eds.) *Connections and Symbols.* Cambridge, MA: MIT Press.

Gabbay, D. (1985): "Theoretical foundations for non-monotonic reasoning in expert systems," in *Logic and Models of Concurrent Systems*, K. Apt ed., Berlin: Springer-Verlag.

Gärdenfors P. (1984): "The dynamics of belief as a basis for logic," *British Journal for the Philosophy of Science 35*, 1–10.

Gärdenfors, P. (1985): "Propositional logic based on the dynamics of belief," *Journal of Symbolic Logic 50,* 390–394.

Gärdenfors, P. (1988): *Knowledge in Flux: Modeling the Dynamics of Epistemic States,* Cambridge, MA: The MIT Press, Bradford Books.

Gärdenfors, P. (1992): "Three levels of inductive inference," to appear in the *Proceedings of the 9th International Congress of Logic, Methodology, and Philosophy of Science*, Amsterdam: North-Holland.

Gärdenfors, P. and D. Makinson. (1993): "Nonmonotonic inferences based on expectations," to appear in *Artificial Intelligence*.

Groenendijk, J. and M. Stokhof (1990): "Dynamic Montague grammar," to appear in *Proceedings of the Second Symposium on Logic and Language,* Hajduszoboszlo, Hungary.

Groenendijk, J. and M. Stokhof (1991): "Dynamic predicate logic," *Linguistics and Philosophy 14*, 39–100.

Grossberg, S. (1989): "Nonlinear neural networks: Principles, mechanisms, and architectures," *Neural Networks 1*, 17–66.

Kraus, S., D. Lehmann, and M. Magidor, (1991), "Nonmonotonic reasoning, preferential models and cumulative logics," *Artificial Intelligence 44*, 167–207.

Makinson, D. (1993): "General patterns in nonmonotonic reasoning," to appear as Chapter 2 of *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume II: Non-Monotonic and Uncertain Reasoning*. Oxford: Oxford University Press.

Makinson, D. and P. Gärdenfors (1991): "Relations between the logic of theory change and nonmonotonic logic," pp. 185–205 in *The Logic of Theory Change,* ed. by A. Fuhrmann and M. Morreau, Springer-Verlag. Also pp. 7–27 in *Proceedings of the Workshop on Nonmonotonic Reasoning, GMD 1990* (Arbeitspapiere der GMD 443), ed. by G. Brewka and H. Freitag, Gesellschaft für Mathematik und Datenverarbeitung MBH, 1990.

Pylyshyn, Z. (1984): *Computation and Cognition*, Cambridge, MA: MIT Press.

de Rijke, M. (1993): "Meeting some neighbours", in J. van Eijck and A. Visser (eds.) *Logic and Information Flow*.

Rumelhart, D. E. and J. L. McClelland (1986): Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Cambridge, MA: MIT Press.

Rumelhart, D. E., P. Smolensky, J. L. McClelland and G. E. Hinton (1986): "Schemata and sequential thought processes in PDP models," in Rumelhart, D. E., *Parallel Distributed Processing*, *Vol. 2*, pp. 7–57, Cambridge, MA: MIT Press.

Sellars, W. (1980): "Inference and meaning," in J. Sicha ed. *Pure Pragmatics and Possible Worlds*, Reseda, CA: Ridgeview Publishing Co.

Smolensky, P. (1986): "Information processing in dynamical systems: foundations of harmony theory," in Rumelhart, D.E., *Parallel Distributed Processing*, *Vol. 1*, 194–281, Cambridge, MA: MIT Press.

Smolensky, P. (1988): "On the proper treatment of connectionism," *Behavioral and Brain Sciences 11*, 1–23.

Smolenskty, P. (1991a): "Connectionism, constituency, and the language of thought," in Loewer, B. and Rey G. (eds.), *Meaning in Mind: Fodor and His Critics*, Oxford: Blackwell, 201–227.

Smolensky, P. (1991b): "Tensor product variable binding and the representation of symbolic structures in connectionist systems," *Artificial Intelligence 46*, 159–216.

Thom, R. (1972): *Stabilité Structurelle et Morphogénèse*, New York: Benjamin.

Veltman, F. (1991): "Defaults in update semantics," in M. Moens (ed), *Common sense entailment and update semantics*, DYANA-deliverable R2.5.C. Edinburgh, 1991, 2–60, to appear in *Journal of Philosophical Logic*.

Woodfield, A. and Morton, A. (1988): "The reality of the symbolic and subsymbolic systems," *Behavioral and Brain Sciences 11*, 58.

Zornetzer, S. F., J. L Davis and C. Lau (1990): *An Introduction to Neural and Electronic Networks*, San Diego: Academic Press.