Fachbereich Informatik der Universität Hamburg

Vogt-Kölln-Str. 30 ◊ D-22527 Hamburg / Germany

University of Hamburg - Computer Science Department

Bericht Nr. 166 • Report No. 166

Computationalism reconsidered
Connectionism and the use of Computer Science concepts in explanations of the mind

Peter Schefe

FBI-HH-B-166/93

December 1993

In die Reihe der Berichte des Fachbereichs Informatik aufgenommen durch

Accepted for Publication in the Report Series of the Department of Computer Science by

Prof. Dr. K. von der Heide, Prof. Dr. M. Jantzen, and Prof. Dr. W. Menzel

Computationalism reconsidered
Connectionism and the use of Computer Science concepts in explanations of the mind

Peter Schefe
Fachbereich Informatik, Universität Hamburg, Vogt-Kölln-Str. 30, D 22527 Hamburg, Germany

Abstract

Computer science concepts, such as symbol, program, implementation, level etc. play an important role in many explanations of the mind. Especially, in the continuing discussion aroused by J. Searle's "Chinese Room Argument", these are (mis)interpreted in conflicting ways, even among computer scientists. After explaining the fundamental computational concepts in question, with special attention to the notion of 'level', different sorts of models in computer simulation are introduced, which differ from each other as to their epistemological and semantic status. Connectionists Networks, as models, turn out to have two incompatible interpretations, "subsymbolic" and "computational". The alleged philosophical implications, as to eliminative materialism and functionalism in particular, are reviewed. In the concluding paragraph, the purported explanations are put into a broader epistemological context. Whilst bodily processes are apt to be partially explained "computationally", the first person phenomenon of the conscious mind does resist these third person explanation attempts.

Keywords. Connectionism, computationalism, functionalism, eliminative materialism, computational modelling, causal explanation, functional explanation

> The concepts of computer science provide the crutches of imagination we need if we are to stumble across the *terra incognita* between our phenome-nology as we know it by "introspection" and our brains as science reveals them to us.
>
> *Daniel C. Dennett*

1 Introduction

In a recent paper on the philosophical implications of connectionism, Lycan[1] writes:

"Neither living things nor even computers are split into a purely "structural" level of biological/physiochemical description and any one "abstract" computational level of machine/psychological description. Rather, they are all hierarchically organized at many levels, each level abstract with respect to those beneath it but structural or concrete as it realizes those levels above it. HF (Homuncular Functionalism – P.S.) allows us rightly to see the functional/structural or software/hardware distinction as entirely relative to a chosen level of organization."

This is an example of how computer science concepts are used in explanations of the mind. It is the notion of *level* that plays a crucial role in Lycan's *philosophical explanation*. There have been several attempts of philosophers to refute this use of computational terms in explanations of mind or brain phenomena, the most prominent being Dreyfus[2] and Searle[3]. Connectionism, even somehow appreciated by the critics mentioned[4], now revives the discussion. It turns out that most of the computational terms used in Symbolic AI are are taken up again in the discussion of connectionism. Chalmers, e.g., claims that Searle's Chinese Room Argument (CRA) "does not go through"[5] as to connectionist systems, resorting to the notion of level as well:

"In a connectionist system, on the other hand, the computational and representational levels are quite separate [...] there being semantic content at the level of the distributed representation [...]"

It is difficult to interpret this statement appropriately. The notion of level is not only interpreted differently in different philosophical contexts, e.g. in "level of description", "ontological level" etc. but also in different disciplines. Computer scientists often try to apply their concepts of *technological*

---

1 [Lycan 1991]
2 [Dreyfus 1972]
3 [Searle 1980]
4 [Searle 1992], [Dreyfus 1992]
5 [Chalmers 1991]

explanation to phenomena of mind. Some are even reproaching philosophers of not being aware of technical concepts used in their arguments, e.g. Perlis[6]:

"He [Searle, P.S.] seems not to understand virtual levels in computational systems at all."

whilst, at the same opportunity, Hayes appears to be more careful:

"I am currently working on a response to Searle which has this very theme: how Searle's failure to understand the concept of levels of interpretation (among others, notably that of the causal story to be told about software) has misled him. I think we should acknowledge, however, that we don't fully understand all this stuff ourselves."

On the other hand, Searle[7] has some complaints:

"Even, more amazingly, a lot of very technical sounding notions are poorly defined – notions such as "computer", "computation", "program", and "symbol", for example."

Searle is not right with respect to technical definitions in theoretical computer science. These are very precise. The problem addressed is, however, what sort of *explanations* computer science concepts can provide especially when applied to mental phenomena. Antony[8], e. g., "conceives of" concepts "for his purposes":

"Functional architectures, virtual machines, and programming languages, accordingly, can be taken as roughly equivalent, and should be contrasted with the algorithms or programs that get executed in them."

What does "roughly equivalent" mean? Are "algorithm" and "program" synonyms? Are programs different from virtual machines? etc. Should one observe the principle of charity in such discourse across disciplines? I think not in general. *Computation* and related concepts have been defined and explained in its home discipline, computer science. If philosophers are trying to gain from their use, they should resort to these explanations.

---

6 on a recent "Virtual Symposium on Virtual Mind",trying to refute Searle's CRA

7 [Searle 1992]

8 [Antony 1991]

First, I will try to explain mostly informally the fundamental concepts in question such as *algorithm, program, symbol, implementation, level.* Secondly, it seems necessary to give an account of different notions of *explanation* in different sciences in order to clarify the explanatory power of a notion used across different disciplines.

The main theses and arguments are the following. Computational concepts are *abstract.* Therefore, chains of computational states do not exhibit any causal relationship, and moreover, it is not intrinsic to any real (concrete) device to be in a computational state. Especially, the computational notion of *level* does not allow for inferences pertaining to the *ontology* of cognitive systems. Computation cannot account for intentionality, as there are no *emergent* properties in computational systems. As any computer model of some domain exhibits a three-place relationship (program - model - domain) it follows that there is a twofold semantics, a computational and a domain-oriented one. Symbolic AI fails to make this distinction. Connectionism is even mistaken in contending a non-symbolic (continuous) device to compute, what turns out to be a contradictio in adjecto. It is overlooked that so-called subsymbolic computation is parasitic on the genuine concept of (symbolic) computation. Hence, both eliminativists and functionalists will neither gain nor be threatened by Connectionism. Finally, computational explanations are discussed in a broader epistemological context. According to the philosopher of science M. Heidelberger, computer science explanations are twofold, functional(technical) and mathematical, i.e., we may use computational terms to explain physical behaviour, but that is not the whole story. There is no way of explaining first person phenomena of the intentional, conscious mind in computational terms at all.

2. Fundamental computational notions

2.1 Technical aspects

The concept of *symbol* is related to other computational concepts – algorithm, program, interpretation, implementation, to name the most important ones. We have to start with the most fundamental:

An *algorithm* is a precisely determined *abstract procedure* using basic or primitive *abstract operations* on *abstract objects*.

Example: concatenation of two (linear) lists:

```
The concatenation of list1 and list2 is:
 If list1 is empty then list2
 else the list of
  the first element of list1
  and the concatenation of
   the rest of list1
   and list2
```

This *recursive* algorithm given as a functional (LISP–like) *program* (see below) uses the primitive operations:

```
is empty:          list      -> {true, false}
list of:           any object x list    -> list
first element of:  (non-empty)list   -> any-object
rest of:           (non-empty)list   -> list
```

These primitive operations (recognition, construction, selection) also constitute the *sort* of the *abstract objects* called "lists". Thus, abstract objects and processes are entities that cannot identified in time and space. Instead, they are specified by linguistic expressions[9]. In computer science, these are expressions in a formal language. Abstract objects may exhibit space-time-relationships such as adjacency, sequence etc. However, these are abstract, too.

Programs (see below for a more precise account) are *descriptions* of algorithms in a formal language.

Everyday life procedures that are sufficiently routine or recipes are apt to to be *described as if* they be algorithms. Standardized and deprived of their

---

[9] Compare, e.g., [Tugendhat/Wolf 1983]

concrete meaning, they may serve as examples of algorithms, e.g., the algorithm of "searching a maze" given below in a Prolog–like style (i.e. , assuming the facts being provided, and backtracking on failure):

Example: searching a maze for room A, starting in some room X:

```
If you are in room A (A = X) then success!
Else if you have been already in room X
     or there are no more doors left to try
     then failure!
     else try next door, call the room it leads to X and
      start searching once more!
```

Remember that this procedure is quite abstract: there are no concrete doors or rooms involved whatsoever. A and X are *symbols referring* to objects of some abstract sort "room". Likewise, there is no "searching" in the sense of an intentional goal–seeking activity. Instead, an abstract relation instance is computed; that is, an X is determined such that there exists a tuple <A, X> that is in the abstract relation at hand.

An algorithm does exhibit neither teleological nor causal relationships. These are intentional interpretations ascribed to it when used as an abstraction of some concrete procedure. An algorithm is an abstraction of real procedures to an abstract sequence of operations on abstract objects.

A *computation* is any abstract execution of an abstract procedure (= algorithm) operating on abstract objects (also called "symbols" or "symbolic structures").

"*Symbol*" is a term often used misleadingly both in computer science and in cognitive science. Basically, it should be replaced by the term *atomic*

object"[10] which is a given abstract individual "thing" without any internal structure. Objects of this primitive kind are *tokens* always being of some *type*. Atomic abstract objects – by definition – cannot be decomposed but are considered to be different from each other. Different *occurrences* or *tokens* of abstract objects of this kind can thus be tested whether they are of equal type or not. These objects can be taken into aggregate objects such as pairs, lists, trees or other *symbolic structures*.

What makes things more intricate: abstract objects can be used to *denote* other abstract objects, such that the notion of *symbol* makes sense. *Denotation* is an unambigous abstract three–place relationship between an *object* denoting or referring to another *object* for an abstract *interpreter* or user (= a program). It allows for *access* to the denoted object by the process of formal *interpretation*, especially *dereferencing* the denoting object. Summarizing this brief account of the computer science notion of symbol:

A *symbol* is an abstract, atomic, identifiable individual *object* that can be *used or interpreted* to formally *denote* , unambiguously, and precisely, any other abstract *object*[11].

Clearly, the explanation given above is not independent of an explanation of natural language symbols. Although natural language symbols and denotations differ from programming language symbols and denotations in being informal, ambiguous, imprecise, and context dependent, all of them are *intentional*. Even if we formalize our intuitive notions we do not escape intentionality or *aboutness*, when using formal schemes for specification *of something* . The notion of *interpretation* turns out to be ambiguous. There is an inverse relationship to the notion of *implementation*. Hence, it can be expected to be ambiguous, too. Thus it should not be astounding that computer scientists are arguing about its intuitive meaning[12]. The notion of implementation appears to be among the philosophically most interesting concepts of computer science. It seems to explain how the mind materializes.

---

[10] See also [Chalmers 1992]

[11] We will neglect the the difference between constants and variables here

[12] Perlis and Hayes in [Hayes et al. 1992]

Implementation is also addressed as *realization, concretization or representation* of an algorithm in the *causal* processes of a purposive technical arrangement of matter called a *machine*. However, this is not the only meaning. In his famous article, Turing[13] writes:

"Strictly speaking, there are no such machines. Everything really moves continuously. But there are many kinds of machine which can be profitably *thought of* as being discrete-state machines." (his italics, P.S.)

Turing has shown - which is a commonplace now in computer science - that for every algorithm there exists an automaton carrying out just this algorithm. This is an *equivalence* relationship. Thus, the notion of an *abstract* implementation becomes feasible: an automaton can be *programmable*. A *program* is the *simulation* of a mostly specialized machine on a universal machine (in general). Such simulated machine is also called a *virtual machine*. To summarize:

A *program* is an unambiguous, precise, symbolic *description* of some algorithm in relation to some *interpreting* machine. It implements an abstract, virtual machine.

This is an abstract relationship involving two types of abstract objects, namely algorithms or automata. The *semantics* of a program is, then, provided by the way the interpreting machine interprets it (which can also be specified by a logical characterization). We may call this notion of semantics *semantics 1*.

It is important to notice that the abstract procedure or algorithm and its concrete realization or virtual implementation are intentionally related to each other. As we indicated for symbols above, this interpretation is informal; we are saying *what the implementation is about*: the physical entity at hand is *intended* to be a machine, to be the realization of some abstract concept of algorithm, such that every orderly behaviour of this device can be *interpreted*

---

13 [Turing 1950]

as if performing some computation[14]. In other words, the physical entity does not compute at all. Computation is an abstract process[15].

Implementation gives rise to the also ubiquitous notion of *level*[16] often exploited in strong AI, functionalism, instrumentalism[17], and homuncular functionalism. As Lycan[18] puts it:

"In any case, psychological, biological, and mechanical systems alike are hierarchically organized, often on the principle of what computer scientists call '*hierarchical control*'." (my italics, P.S.)

What kinds of hierarchies do hardware/software systems exhibit satisfying these requirements? There are at least two candidates. First, there is the *simulation hierarchy*. The basic level is understood to be the physically realized machine (in principle a universal machine providing , say, some logical and arithmetic operations on fixed-length strings of Boolean objects or numbers, respectively). "On top of" this machine, a virtual machine can be build as explained above. This sort of engineering can be repeated to an arbitrary number of levels. In particular, every universal machine can simulate itself *ad inifinitum*.

The second one is established by *levels of procedural nesting*[19]. For instance, in order to carry out a multiplication, addition is used, and to carry out addition, the primitive operation of successor is called. Every such procedure can be viewed as a *black box* with a given *functionality*. All these operations are on the *same implementational level*: Only the *primitive* operations have to be implemented by a lower machine. (This is good engineering). However, the notion of level here becomes more and more

---

14 See also [Searle 1990]

15 That may even be hard to swallow for computer scientists. We will find below that this is reflected in the two kinds of explanations computer science can provide (See [Heidelberger 1993]). Fetzer ([Fetzer 1991]) argues, accordingly, that the mathematics of program verification cannot account for the correctness of programs running on a concrete machine.

16 See [Hayes et al. 1992]

17 [Dennett 1978]

18 [Lycan 1991]

19 We are neglecting here the difference between static and dynamic embedding.

blurred, if it is taken into consideration that nesting normally is not strictly hierarchical, e.g. , in recursive or mutually recursive procedures.

2.2 Philosophical interpretations

The philosophical implications should be rather obvious now: computational notions do not support any solution to the mind-body problem. To put it into a nutshell: computational notions are abstract – mind and body are concrete entities. Hence Fodor 's "definitions":

"Computations just *are* processes in which representations have their causal consequences in virtue of their form"[20]

"A computation is a causal chain of computer states..."[21]

exhibit a deep misconception. Only physical processes (physical machine processes) can have causal effects. These are not *identical* [22] to abstract ones. A simple example may serve as a demonstration: In a counting machine for coins, each falling of a coin "causes" a state change, e.g., the turning of a wheel, say, but there are no causal relationships between numbers "represented by" the different machine states. A number is abstractly "generated" by the application of a successor function to its predecessor, not by any physical process. Conversely, the functional specification of the hardware does not require any reference to numbers.

It is our interpretation and intentional use that makes physical entities into computers. Realization is a teleological, not a naturalizable concept. Fodor's *is* is not warranted; abstract entities cannot be identical to concrete ones. Only concrete physical states can be said to make up a causal chain. As identity is a transitive relationship, we would have to expect the absurd consequence that different realizations are identical. Of course, they are *equivalent under our purposive interpretation and use.*

---

20 [Fodor 1981b], p. 325

21 [Fodor 1981a], p. 131

22 What is at stake here is normally called token identity theory. This theory is false, because concepts abstracted from concrete processes cannot be identical with these. Compare [Keil 1993]

As to the notions of level and hierarchical organization, a considerable contribution to the prevailing confusion in both computer science and cognitive science is due to Marr's "three levels"[23], "computational, algorithmic-representational, implementational". Firstly, there is *no physical ("implementational") level* at all: the lowest level is the abstract machine that is physically realized. This machine is usually "thought of" (Turing) as a universal automaton. Thus, secondly, there are not just two distinct levels – "computational, algorithmic" – that are different from each other, but arbitrary many levels which are "algorithmic-computational". It does not make sense to distinguish "algorithmic" from "computational", echoed in Pylyshyn's[24] distinction "functional, intentional". McClamrock[25] who also argues against this three-level-dogma, unfortunately, interprets different levels of implementation as comparable to organizational levels of the brain. Hayes is committed to identy theory as criticised above:

"What it is that makes computers into computers [...] is that they are machines whose behaviour is influenced in systematic ways by the meanings of the symbols that we input to them [...] Now, if we look at how that *is* possible, then there turn out to be, as Perlis correctly emphasizes, layers of interpretation of code on virtual machines of one kind or another (and this is not hermeneutical confusion, by the way, but sound engineering talk)."

Hayes, unvoluntarily, forges the weapon which is to be turned against him: it's all hardware and software *engineering*, i.e. , a purposive, intentional making and *interpretation* of some physical or virtual machine. To understand a machine or a program, you have to take into consideration its creator and her intentions[26].

The level talk does not help. One mistake in Lycan's (and Dennett's) argument is the contention that there be lower "degrees" of intelligence on the lower levels, and, eventually, there will be one of "degree" zero, a pure "machine". Although the programs arrived at thereby may become more and more powerful or complex in the sense that a bulk of first–level operations

---

23 [Marr 1982], compare [McClamrock 1991]

24 [Pylyshyn 1984]

25 [McClamrock 1991]

26 See also [Margolis 1980], and [Keil 1993] in particular

may be carried out to account for one step of the top level program, there is no sense in talking of more "intelligent" programs on higher levels and more "stupid" ones on lower levels. As Lycan[27] states the contrary position:

"Attneave's original breakdown strategy of avoiding the standard regress objection to homuncular explanation also answers the deep metaphysical question of how *intelligence* (amazing degrees of it, in some subjects) can emerge from ontologically just a great mass of entirely insentient, nonintelligent molecules. For a homuncular breakdown analyzes intelligent beings ultimately into sub-sub "agencies" any of which, in Dennett's phrase, 'can be replaced by a machine.'"

That he has in mind the hierarchy of procedural nesting is obvious from his explanation:

"Thus an organism's complete psychological description would consist of a flowchart depicting the person's immediately subpersonal homunculi or agencies and their routes of cooperative access to each other, followed by a set of lower level flowcharts [...] and so on. At any given level, the flowcharts show how the components depicted at that level cooperate to realize the capacities of the single agency whose functional annalysis they cooperatively constitute."

It is hard to see how this inflation of levels or agencies can be kept from running into infinity. There is no distinct stopping level as every agency (even the extremely "stupid" ones) is still an agency, not a machine[28]. Lycan does not realize that every identifiable level constitutes a virtual machine, and, in more technical terms, that the high-level operations are transitively related to the lower-level ones by the *same* kind of *abstract* implementation-/interpretation relationship[29]. There is only a *definitional, not an ontological reduction*. Why should a LISP machine implemented on top of a Prolog machine exhibit "higher mental life" than that on its bottom? Why should a multiplication function be more "intelligent" than one that does addition? Of course, the lowest (by definition) machine is the only one physically implemented, but each level's machine could be as well.

---

27 [Lycan 1991]

28 Compare Keil's lucid critique of homuncularism in [Keil 1993]

29 The lower level entities and their associated procedures , e.g., lists, may be interpreted as, say, numbers on the higher level. Numbers can be implemented by lists.

It is also this purported ontological physical/mental distinction in real computing machinery that drives Hayes to insist that the lowest level must be a physical machine working according to the laws of physics[30]. What else is technology than exploiting the laws of nature to some purpose? In order to put software to work without human operation, it has to be realized as *physical data* for a physical machine. Thus all operations on higher levels can be causally explained in the same way as those on lower levels. In other words, there are *no emergent properties*[31].

3 Computational modelling

Computational modelling, also called computer simulation, involves the *use of models as substitutes for reality,* mostly because the domain modelled is not accessible for direct exploration or an analytical model doesn't avail. Simulation models therefore cannot provide explanations as analytical models do, but only hypotheses. Hence, a computer simulation is a *three-place intentional relationship* between a *computer program* describing an *abstract model* accounting for a *domain*. The program thereby gets another interpretation than just an algorithm. On a broader scale, a model may be:

(1) a concrete physical analog substitute of the original, especially, analog "computer" (example: model aircraft in wind channel)

(2) a mathematical characterization (a set of differential equations) describing continuous space-time behaviour of the original (example: physical device model)

(3) a logical characterization (a set of predicate formulae, a theory) describing the discrete possible state of affairs pertaining to the original or its input-output-behaviour or functioning or... (example: digital device model)

---

30 Hayes worries about Searle's claim of *being* an implementation of some rule interpreting machine, because he doesn't acknowledge the intentionality of the implementation relation

31 *Emergent* is a property of a system if it cannot be predicted given the properties of the components.

(4) an algorithmic characterization (an automaton or a program for an automaton) describing the discrete behaviour of a domain entitity (example: queue model)

These models differ as to their epistemological status. Where do computational models of the mind fit in?

I should not elaborate on (1) so much; however, computationalists may treat a robot, say, as an analogue of an human being[32]. This way of using models instead of originals may lead to an equivalence presupposition, as the laws governing the model are, or are considered to be, the same as those governing the modellee. I doubt that it makes much sense to call this relationship "modelling", because abstraction is missing. Are two cars built according to the same prototype models of each other? Are twins models of each other? Because certain laws holding for both of them enables the observer to use either of them to learn about the other? We will come back to this issue when discussing connectionist claims.

(2) through (4) are abstract models in the sense that they apply some equivalence relation to subsume different concrete occurrences of natural phenomena under the same concept. (2) is the basis for lawful explanation in natural science, but may be used as a basis for computer simulations also, if the real domain is not accessible. Simulations of this sort are always thought of as *derived or approximative*. Examples for the latter may be thunderstorm models or system dynamics world models. They do not give rise to a new philosophical problem.

Most interesting and confusing are (3) and (4), which are closely related to each other. Because of being *discrete* or symbolic descriptions they are programmable in a more direct way. As modelling is a three–place relationship, such a program has two meanings:

– the abstract implementation of an algorithm on a given concrete or virtual machine (semantics I)

---

32 See, e.g., [Tetens 1993]. Tetens conceives of a Gedankenexperiment, in which robots can be damaged and learn to express pain when treated as fellows in a society of human beings.

– *functioning* as an abstract model of a concrete or abstract domain (semantics II)

3.1 Symbolic AI

Before considering the alleged "revolutionary support" connectionist models lend to certain philosophical conceptions of the mind[33], I will shortly address the "classical" debate concerning Symbolic AI. Why does its supposition that a mind cannot only be modelled but be built into a computer still prevail? Why is this view so resistent to arguments such as Searle's CRA?

Whilst there is no temptation of taking implemented models of thunderstorms to be thunderstorms themselves[34], it is the metaphorical power of the Turing Test – and Searle's CRA is on this strand – that renders this viable for cognitive simulations: a computer program that mimics intelligent behaviour entirely *must be* intelligent. Searle tried to refute this contention by pointing to its still purely "syntactic" character thereby implicitly addressing what I dubbed *semantics I*. It is not necessary to review here all positions adopted in response to this argument. The Systems Reply and The Robot Reply appear to be well known[35].

Why, then, did CRA not convince Perlis, e. g., who says:

"What the Computational Thesis (CT) posits is precisely that it is a functional level of activity brought about by mundane nonmental actions (neurons, circuits, whatever)."

As an epistemological consequence of this thesis, Perlis must interpret the CRA as question begging. He has to require that the CRA rules out that the Room may "instantiate"[36] a virtual mind. Even if Searle denies

---

33 [Ramsay et al 1991]

34 Dreyfus [Dreyfus 1972] already pointed to this attitude twenty years ago in order to argue against it.

35 See [Searle 1980] for summaries of these objections

36 Compare [Antony 1991] for an explanation of the difference of "instantiation" and "implementation"

understanding Chinese (what Perlis acknowledges), he may still "implement" a Chinese understanding virtual machine. Hence:

"To take the internalized Chinese Room as evidence against (the levels version of) CT is to have understood neither the CT nor virtual levels in computers."

Searle's standard answer to the so called Systems Reply – to memorize all the rules and do all the processing in his head – cannot change this belief. He may not be conscious of his ability, according to one of the responses we need not consider here in detail. Especially, Dennett is not yet prepared to follow Searle's gedankenexperiment, but is counting on "Mother Nature" *who* lets intelligent behaviour *emerge* , as the software will be sufficiently complex, and blaming opponents for their lack of "imagination".

The second main counter-argument to the CRA, the so called Robot Reply, is revived by Connectionists Networks in particular, and therefore will be dealt with in the following section.

It is one of the shortcomings of the CRA that it concedes too much; its science fiction character only obscures the issue Searle has in mind. Perlis is right: the CRA is an argument pro AI, not contra AI. It is a pity that people spent so much time worrying about this argument[37] . In a more recent paper[38], Searle argued that even syntax is not intrinsic to physics, admitting at the same time:

"This is a different argument from the Chinese Room Argument and I should have seen it ten years ago but did not."

---

37 To immunize the Turing Test (TT) against the CRA, Harnad has invented a "Total Turing Test" (TTT) comprising the test of bodily behaviour and capabilities by endowing the machine with sensors and effectors. (See [Harnad 1991b]). It is obvious, however, that this would not be necessary, as there can be no successful linguistic imitation of any human being without world knowledge that can be only acquired via "sensor/effector" interaction with the world.

38 [Searle 1990b], p. 594

3.2 Connectionist AI

Connectionist systems are claimed to be a new *paradigm* in cognitive science[39], or, as some put it, cognitive neurobiology[40], and to overcome the deficiencies both of Symbolic AI and most of the conceptions of philosophy of mind. Even Searle, one of the main critics of Symbolic AI, appears to be impressed:

"Among their other merits, at least some connectionist models show how a system might convert a meaningful input into a meaningful output without any rules, principles, inferences or other sorts of meaningful phenomena in between."

We are not going here to present the technical details or report on the relative merits of different approaches within this "paradigm"[41]. Churchland gives an interesting summary:

"All told, this network is a device for transforming any one of a great many possible input vectors, (i.e., activation patterns) into a uniquely corresponding output vector. It is a device for computing a specific function, and which function it computes is fixed by the global configuration of its synaptic weights."[42]

Notice that a network is viewed as a *device*, i.e., a *purposive* arrangement of matter, that it is *said* to *compute*, i.e., to perform an *abstract* process. An example often mentioned is NETtalk[43] that learns to pronounce English text. By a hill-climbing learning sequence, the system adjusts its numerical transformation device to establish the required mapping from graphemes coded as input vectors to sound effection patterns coded in the output vector approximately correctly.

---

39 E.g., [Smolenky 1990]

40 [Churchland 1988]

41 See, a. o., [Rumelhart/McClelland 1986], [Bechtel/Abrahamsen 1991]. For a critical review of different techniques developed so far, see [Dreyfus 1992], who takes a phenomenological stance, whilst I am going to pursue a more analytical line of criticism.

42 [Churchland 1992], p.202

43 [Rosenberg/Sejnowski 1987]

3.2.1 The basic assumptions

As we are going to discuss the purported philosophical implications of connectionist networks (CNs), we have to assess the notion of a CN, especially its alleged distinctiveness from conventional computing systems such as Turing Machines (TMs).

CNs are said to be to *parallel* and *distributed*[44]. There is no special problem with these notions. Parallel and distributed computational systems can always be simulated (= abstractly implemented) as virtual systems on sequential TMs. The architecture may offer advantages as to behaviour in time and space, but there is no such system that could not be proved equivalent to some (possibly restricted) TM[45].

However, there seems to be more about it. To get through the labyrinth of ideological presuppositions as to CNs is even harder than through the ideological halo pertaining to TMs. Smolensky[46], one of the leading ideological figures, tells us:

"I will now argue that these models should be viewed as discrete simulations of an underlying continuous model, considering first discretization of time and then discretization of units' values."

CNs are understood implicitly (even if Smolensky may deny this) as a mathematical model of the behaviour of continuous *physical entities*, perhaps brain–like systems, in terms of physical input–output–relationships. So there are two modelling relationships: the simulation uses a model of type (2)[47]. The modellee, in this relationship, in turn, is a model of type (1): the presupposed artificially arranged physical entities are used as substitutes for real cognitive systems. From our above argument, it follows that, in this system, there will be

---

44 [van Gelder 1992] defines: "[...] a representation is genuinely distributed if –roughly– it is representing many items using exactly the same resources." inspired by the clinical phenomenon of prosopagnosia, the inability to recognize faces, which always appears as a total, not partial, loss of that ability.

45 See [Adam et al. 1992] for a comparison of CNs and TMs, and [Schwarz 1992]

46 [Smolensky 1990]

47 This is obvious from a formulation of [Bechtel/Abrahamsen], p. XIII: "Connectionists networks are dynamical systems that are described by mathematical equations."

no computation going on at all, as there is no computation going on in stones falling to earth. However, Smolensky concludes:

"The final point is a foundational one. The theory of discrete computation is quite well understood. If there is any new theory of computation implicit in the subsymbolic approach, it is likely to be a result of a fundamentally different, continuous formulation of computation."

It is really hard to see what the meaning of this notion of "computation" would be like[48]. *Computation is symbolic*[49]. Smolensky would challenge the classical thesis of Church and Turing, saying that for every computation there will be some Turing-machine that performs it; only a thesis, but quite plausible. A corrolary thesis is that processes that cannot be dealt with algorithmically, i.e., symbolically, are not programmable on any TM. A classical example are the real numbers which are not computable or are not "given" to any TM. However, there are *useful approximations*[50]. Simulations of *continuous* phenomena, therefore, always use models of type (2). To emphasize, what Smolensky addresses is not computation, but a presupposed functioning of a physical system. This functioning is simulated on a digital computer or TM. The simulation program, then, has a twofold semantics as explained above:

---

[48] Smolensky's contention is echoed, e. g., in [Bechtel/Abrahamsen], p. 3: "The connectionist view of computation is quite different. It focuses on causal processes by which units excite and inhibit each other and does not provide either for stored symbols or rules that govern their manipulations." Antony seems to be aware of the problem: "[...]there are serious difficulties with the current understanding of Connectionist computation", [Antony 1991], p. 324

[49] This is not legislation. It is Smolensky's turn to take the onus of proof for his "new theory of computation". The paradigm of symbolic computation established by Turing, Church , Post, Kleene a.o. cannot be abandoned by plain appeal.

[50] A related issue is the approximate computation of *random numbers* that are essential to *discrete* simulations. Randomness cannot be dealt with algorithmically in principle; all random number generators are biased, but often may serve as a *sufficient* approximation. To introduce randomness into a simulation, you have to establish a link to the "outer world" drawing upon real random processes. The behaviour in time of a such a system, a computer linked to the real world, can no longer be regarded as the execution of an algorithm. (Otherwise every process would be considered to be the execution of some algorithm)

- numerical computation[51] (semantics I)

- physical continuous functioning (semantics II)

The consequences so far are the following. If the processes modelled are continuous, it makes no sense to consider the processes themselves to be computational, i.e., their implemented discrete model as being equivalent to the modellee. If the processes involved *can be thought of as* computational, it is in the eye of the beholder or user. This, however, renders the concept vulnerable to criticism as presented above.

What about the mysterious notion of "subsymbolic"? Smolensky, when saying:

"The claim here is that the most analytically powerful descriptions of subsymbolic models are continuous ones while those of symbolic models are not."

uses the term as a synonym for "continuous", thus disguising its conceptual emptiness. He then relates CNs to "analog computers", not being aware of the fact that these devices are "computational" only in a parasitic way, by using physical measuring techniques, getting scale values into this device, and, after some physical operation, reading off some value, prone to error in a specific way. Only in this parasitic way, can CNs be said to "compute" a function by associating certain scale value vectors as input with certain scale value vectors as output. Of course, the working of a slide rule, a simple analog device, is physical–continuous. Whether it's said to do addition or multiplication depends on *our* discrete–valued input/output scale interpretation. It makes no sense to call the continuous states of internal or external units of analog devices "symbolic", however, and "subsymbolic" makes no sense either. They may be interpreted as *quantities*. Only by an abstraction process in the eye of the beholder can these be interpreted as *codings*[52] of symbolic entities. The "symbolic, subsymbolic" distinction is also related to the misconception of

---

51 which is usually abstractly implemented on a symbolic machine

52 By the way, "coding" is a computational term naming a mapping between symbol systems. There is only a metaphorical talk about physical coding, neural coding etc.

levels in software engineering. One possible interpretation adopted, e. g., by Clapin[53], is to map it onto the "algorithmic, implementational" distinction or to take the CN as an analogon to the von Neumann CPU, by simple *causal* processes. We could leave it here. As the conceptual confusion appears to be a spreading activation process, however, I would like to address some of their typical instances.

3.2.2 Philosophical interpretations

Connectionism, as an ideology, comes in mainly two flavours, eliminative-materialist, and functionalist[54]. Eliminative materialists regard the alleged computational properties of CNs as a new justification for their way of escaping the mind–body problem. Alternatively, in order to maintain computationalism, they abandon the mental altogether. I am not going to refute this quite implausible position at length[55]A short discussion of eliminative materialist trying to refute the *self application argument* may be sufficient. This argument, also described as *reductio*[56], says, briefly, that the eliminative materialists are just abandoning what they are grounded on, the mental. If somebody has taken this position, however, as, for example, P. Churchand[57] has, the counter–argument appears to him as a petitio principii. This kind of self–entrenchment is sometimes reflected in a kind of revolutionary attitude. Instead of arguing, for example, Ramsay, Stich, and Garon[58], compare the mental to theoretical constructs like phlogiston, simply denying its existence. If they commit themselves to Connectionism, they have

---

53 [Clapin 1991]

54 See also [Christiansen/Chater 1992]

55 For a more thorough discussion, see [Searle 1992] and [Hastedt 1988].

56 [Churchland 1992], p 21f

57 [Churchland 1992]. For the reader not acquainted with the "refutation" of the Churchlands. By inserting "vital spirit" for "meaning" into the argument, they are trying to show its question begging character. Then they get the conclusion: "But if he is dead, then his statement is a meaningless string..." Obviously, this is incoherent, as they forgot to replace "meaning" in "meaningless". It is *impossible* to argue that there is no meaning, because arguing cannot do without meaning.

58 [Ramsay et al. 1992]

to accept just what they are trying to pin on their opponents[59]. Their attempt to object to functionalist claims about CNs is in no way sustaining their own interpretation, as it is no argument *for* a senseless position to argue *against* another senseless position.

Whilst the eliminativists falsely regard CNs as sustaining their denial of mental beliefs, the functionalists interpret them falsely as cognitive devices, "subsymbolic", or "implementing"[60] symbolic ones. The Robot Reply to the CRA:

"that the meaning of the symbols comes from connecting the symbol system to the world"[61]

i. e., that the computer has to be equipped with sensors and effectors which are said to "ground" the symbols, is quite revived now, by turning to Connectionism[62]. However, causal connections do not explain *aboutness*: A world object may , e. g., *cause* a print-like pattern in the robot mediated by a sensory device (as in pictorial analysis systems), which may undergo, sequentially or in parallel, different transformation processes, classifying it according to some preprogrammed or "learned" scheme eventually. All these processes can be explained functionally without any reference to non-physical concepts. Still, there are no emergent properties. Intentionality remains in the eye of the designer or user.

Functionalism is fundamentally mistaken[63] in claiming the multiple implementability of the mental by alluding to the hardware–software distinction[64] of implementations, as we have indicated above: the mental cannot be reduced to computation. That this is an unwarranted illusion, as

---

59 One could direct this "argument" in the opposite direction: The phlogiston–people jumped to conclusions imposing presumptuous interpretations on poor data – just as Ramsay et al. are doing when they base eliminativists claims on still poor CN performances.

60 e.g., [Lycan 1991]

61 [Harnad 1991a], p. 340

62 See also [Bechtel 1993]

63 See also [Searle 1992], and [Hastedt 1988] for a more elaborated discussion.

64 Even in the sense Ramsay is explaining in [Ramsay 1989]

has been stated even by researchers in favour of a functionalist point of view, e. g., Christiansen and Chater:

"Crucially, the distributed representations in question are only non-arbitrary in relation to the structure of the given input representations, not in relation to what the latter are representations of, i.e. the entities they refer to in the outside world. Consequently, similarity is defined as a relation between input representations and not as a relation to the appropriate external objects they are to represent. Furthermore, since the input representations provided by the programmer are typically pre-structured and of a highly abstract nature, it is always possible to give a network's input representations a different interpretation, thus changing the projected content of the internal distributed representations."[65]

The authors seem to recognize the resurrection of the AI fallacy in the CN ideology, namely that intrinsic semantics of the "representations" will emerge somehow. Nevertheless, they still believe in the capability of these devices to bring about essential features of symbolic systems, especially compositionality[66]:

"What is required, it appears, is not a new notion of compositionality, but the attempt to devise networks which can behave as if they had structured representations, followed by an analysis of their workings [...] what kind of compositionality we should ascribe connectionist representations is an empirical question, which can only be answered by empirical investigation."[67]

So there is no departure in principle from AI as "empirical enquiry", as conceived of by Newell and Simon[68]. Instead, there seems to be a view shared among most researchers that, because of the brain-inspired structure, CNs may exhibit *internal semantics*, as expressed, for example, by Chalmers[69]:

---

65 [Christiansen/Chater 1991], p. 233

66 Especially [Goschke/Koppelberg 1992] are concerned with endorsing the CN view of "weak compositionality". Referring to various empirical findings appearing to contradict the principle of strict compositionality sustained by functionalists, they are not aware of the self-defeating character of this strategy: sensivity to the - situational - context cannot be modelled computationally at all (because of an infinite regress). CNs are not situated beings, but machines *designed* to meet given *purposes*. Compare [Dreyfus 1972]

67 [Christiansen/Chater 1991], p. 243

68 [Newell/Simon 1992]

69 [Chalmers 1992], p. 47

"I have argued that if we use representational vehicles that are not primitive tokens, but instead possess rich internal patterns, the problem of intrinsic content might be solved."

Chalmers was "arguing":

"The fact that there is syntactic manipulation going on at the level of the individual node does not stop there being semantic content at the level of the distributed representation any more than the fact that the cells in the human brain obey iron-clad laws of physics stops there being semantic content at the level of the concept."

If this is an argument at all how are syntactic rules and physical laws related to each other such that it is justified to establish a logical equivalence between the contention that there is "semantic content at the level of the distributed representation" and the fact that there is a semantics of concepts?

Dennett being well aware of the difference between *ascribed* rules and *descripitive* laws, nevertheless, evokes "Mother Nature" as a designer that "discovers", by evolution, "*wise* rules":

"Such rules no more need be explicitly represented than do the principles of aerodynamics honored in the design of birds' wings."[70]

There are at least two mistakes in what Dennett tries to construe. First, he is repeating his long-standing error that intentionality of (conscious, purposive) design be present in *natural* processes (or that intentionality could be naturalized[71]). Second, he is misinterpreting CNs as computing devices, sharing a common misunderstanding of CN advocates. Third, he is mistaken in assuming that the physical patterns simulated in CNs are brain-like.

So the most confusing misunderstanding of CNs shared by different schools of interpretation is the identification of model and modellee with respect to being computational. Even abandoning this misconception, we may question further whether CNs are reasonable models of type (2) for the *brain*. There is some evidence that they are not. A. Iran-Nejad and A. Homaifar[72]

---

70 [Dennett 1986] quoted in [Dennett 1991], p. 25

71 For a thorough discussion of naturalization see [Keil 1993]

72 [Iran-Nejad/Homaifar 1991]

have pointed out quite convincingly that CNs based on early perceptron models are still on the associationistic and behavioristic path – as well as Symbolic AI. Neither are CNs really distributed nor is the level talk – subsymbolic vs. symbolic – in any respect reflecting a property of the distributed brain processes[73]. Connectionists as well as Symbolic AI supporters are misled by the computational notion of level. The former, in particular the eliminativists, when trying to reduce mental processes to alleged brain processes are not aware of being still on some mental level using physical *metaphors* such as "activation", "firing" etc. The notion of level only makes sense if understood as a *level of explanation*. I will come back to that below.

As CNs are neither brain models of type (1) – which could be used as research objects instead of the brain – nor brain models of type (2) – which could generate some plausible hypotheses – what, then, will be the possible outcome of the CN/PDP research project? As did Symbolic AI, CN/PDP will provide, or has already provided, some useful *programming tools*[74].

4 Conclusion: What could be *explained* in computational terms?

Cognitive science and even brain science, however, are still aiming at *explanations* of human cognition. Computationalists treat computer programs as explanations. I don't think they are. But what can computer science explain? To get this clear, we have to relate it to the general philosophical questions: What is explanation? What kinds of explanation does science use? Are there special kinds for computer science and cognitive science?[75]

The purpose of explanation is to make our knowledge more *coherent* and *standardized*. Explanations are the better the more they are apt to achieve this. The most common, most esteemed kind of scientific explanation is strict *causal*

---

[73] See [Iran-Nejad/Homaifar 1991] and [Roth 1991] for realistic accounts of contemporary brain science

[74] Compare [Cummins 1991]. Whilst I disagree with his functionalist view, I think he is right in arguing "When you program a computer you are designing a virtual computer. Connectionists do this by programming just like everyone else."

[75] As to computer science, see [Heidelberger 1993]

explanation as used in *classical* physics. Predominantly, it is still considered to be the ideal of explanation. Some appear to consider this the only valid explanation, e.g., for Searle, physical causation – by "iron–clad laws" – is the right way for the explanation of the mind also, when he bluntly contends:

"Mental phenomena are caused by neurophysiological processes in the brain..."

I think Searle is mistaken, if this statement is suggesting that there will be a "causal explanation" of consciousness and intentionality. (Strict) causality is limited to classical physics of macro phenomena. Modern physics has to cope with probabilities and *statistical explanations*, which are no longer causal. So I disagree also with:

"Consciousness is a higher–level or emergent property of the brain in the utterly harmless sense of "higher–level" or "emergent" in which solidity is a higher–level emergent property of H_2O molecules when they are in a lattice structure (ice)..."

By no means is this conjecture warranted by what we know science can explain at all. Modern physics has shown that there is no such simple *microreduction* of complex phenomena as implied by the examples he is using. I think, Searle's brutally simple contention is exposed to criticism of a kind he is often applying to computationalists himself: you can't attack it directly, because it makes no sense. If brain processes are thought to *cause* mental processes, then both processes have to be "physical" (which he presupposes). This cannot mean that they are "physical" in the sense of being dealt with by classical physics. Hence, "cause" cannot be interpreted this way either. Hence, the comparison given by Searle makes no sense.

The story is more complicated than the classical logical empiricist exspected it to be. There would be no chemistry if everything could be reduced to physical causation. Chemical processes need a *morphological* explanation, i.e., the morphological structure of molecules is used to explain their *chemical* behaviour in terms of *dispositions*[76]. These abstractions from physics implies that we are no longer considering processes governed by the laws of

---

[76] Compare [Heidelberger 1993]

physics[77]. We get at a new *level of explanation*. Morphology, in turn, is not sufficient for the explanation of life. Biology, even molecular biology, uses *functional* explanations, e.g., that the heart is pumping blood and the lungs are bringing about the gas exchange in the blood of animals. Functional explanation, the explanation of the functioning of the whole in terms of the functioning of the parts, is also the normal way of *technical* explanation. No wonder that scientists or philosophers have always been tempted to treat living beings as machines. The difference between these two is normally that biological functional explanations are often approximative, or, concerning neurophysiology, hypothetical; technical ones are intentional and are or *should be* well understood.

Although he is using implicitly functional notions like *system* himself, Searle[78] appears to be reluctant to recognize functional explanations as epistemologically acceptable at all:

"The so called functional level is not a separate level at all, but simply one of the causal levels described in terms of our interests. Where artifacts and biological individuals are concerned, our interests are so obvious that they may seem inevitable, and the functional level may seem intrinsic to the system. After all, who would deny, for example, the heart *functions* to pump blood. But remember that when we say the heart functions to pump blood the only facts in question are that the heart does, in fact, pump blood; [...] To put the point bluntly, in addition to its various causal relations the heart does not have any function."

I think this is a somewhat unduly depreciation of functional explanations. One can't escape it anyway; so Searle is mistaken in claiming that:

"Variable secretions of auxin cause plants to turn their leaves to the sun."

be a "mechanical hardware explanation". There is no difference comparing it to "The heart pumps blood" as a function. *Secretion* is a function as well as some other terms used in the example. We simply can't talk about living systems without using functional terms. This should make us cautious with ontological claims.

---

77 See [Hastedt 1988] for are discussion of two notions of "physical". [Keil 1993] is arguing against this contention, in favour of a non-ontological concept of causality.

78 [Searle 1990b], p. 591

So far, we have not used a *computational explanation*. There seem to be some higher functional aspects of machines or living systems that can be explained this way, for example, when we are saying that the structure of certain molecules plays a computational *role* in a *system*, e.g., the genes are *coding* the features of possible descendants. What does that mean? In computer science, there appear to be two kinds of explanation[79]:

- a functional one, e.g., that the opening of one switch causes the closing of another one

- a mathematical or computational one, e.g., that to prove or compute the consequent of an implication you have to prove or compute the antecedents, respectively

As I pointed out above, the concrete implementation is a three-place relationship that relates *intentionally* hardware and software machines. In living systems, then, we use notions like coding, copying, control not only as metaphors[80], but also as higher *functional* terms. Small changes in the "code" may have large *effects* in the dynamic behaviour, e. g. , growth, of the whole system. To interpret bio-chemical processes as *partly* "computational" is no more arbitrary than to interpret them in technical terms. If we take computation this way, it may have its merits. That *chance* (which cannot be computed[81]) is an essential factor in biological evolution seems to be an established fact[82]. Hence, this applies to any attempt of computer generated *Artificial Life*[83]: there will be no *life* whatever without chance. If we are willing to accept this possibility at all, what will be going on cannot be a *simulation* of life, but another evolution of *real* life[84].

The mind does not fit so well into this inventory of explanations. There can be no neurophysiological or technical, in particular, computational

---

79 [Heidelberger 1993]

80 Compare [Keil 1993] who argues that these ascriptions *are* metaphorical, but nevertheless inevitable.

81 See footnote 49

82 See, e.g., [Kuhn 1973] and the resumee given in [Stegmüller 1979]

83 [Langton 1989]

84 See also [Roth 1991a]

explanation of the conscious mind. Searle is right in arguing that consciousness is irreducible; his above quoted contention, however, is in no way an explanation. Neither will it be possible to eliminate mental talk altogether from "neuropsychology", nor is it possible to reduce the first-person view of conscious experience also admitted by Searle to a third-person one. Neurophysiological research will make progress in understanding the sustaining of the overall functioning of the brain by chemical processes; it will proceed to more sophisticated maps of perception and motor control; it may provide with more appropriate means for curing mental diseases etc. *There will be, however, no microreduction in the sense implied by the examples given by Searle.*

It should be clear that "computational" rules (or virtual "machines") are neither laws nor rules that are followed in the sense human beings are following rules consciously but special, higher functional terms that can be used in technical and biological explanations in the restricted sense indicated above. Dennett's "*wise* rules" are only functional descriptions of biological/bodily processes (not behaviour, thus abstracting from intentional content) and, hence, are no more observer-relative than any other functional description.

It makes no sense to contend, then, that the brain *is* a computer, nor is it reasonable to declare the mind to be a program. Although it is a brute "fact" that there *is* no mind without a body in its environment (which is more than a brain), there will be no objectivistic or third person level of explanation of the mind. The neurophysiologist Roth tells us[85]:

"My conclusion is that we cannot do without concepts such as "meaning" and "valuation", i.e., non-physico-chemical concepts, in the description of brain processes. The ontological leap between the neuronal brain machine and the realm of conscious perception considered important by many philosophers would be a leap only in case of (I) the brain machine being describable as a purely neural machine what, as has been indicated above, is impossible, and (II) this machine existing in a world independent of consciousness and separated, then, from the world of consciousness. The brain, accessible as it is for neurobiologists (as well as for everybody else), however, is part of the cognitive world, the world of consciousness, and, hence, not ontologically different from this world. We only get an ontological leap, if we misinterpret

---

85 [Roth 1991b], p.369f, my translation

the propositions of science (including brain science) as propositions pertaining to a world independent of consciousness."

One may not share this idealistic view of knowledge in general. On the other hand, Roth is right in showing that knowledge of the self cannot eliminate first–person concepts. It is only this perspective that does justice to the mind as a cultural, historical, common sense phenomenon. Brain science will contribute to that knowledge as well as "cognitive science", if its explanations take into account the unsurmountable restrictions of self–referentiality for any objectivistic approach.

References

[Adams et al. 1992] F. Adams, K. Aizawa, G. Fuller: "Rules in Programming Languages and Networks", In: [Dinsmore 1992], pp. 49–68

[Antony 1991] M. V. Antony: "Fodor and Pylyshyn on Connectionism" *Mind and Machines* 1 (1991), pp. 321-341

[Bechtel 1988] W. Bechtel: "Philosophy of the Mind. An Overview for Cognitive Science" Lawrence Erlbaum, Hilssdale, NJ, 1988

[Bechtel 1993] W. Bechtel: "Currents in Connectionism" *Mind and Machines* 1 (1993), pp. 125-153

[Bechtel/Abrahamsen 1991] W. Bechtel, A. Abrahamsen: "Connectionism and the mind", Basil Blackwell, Cambridge, Mass. , 1991

[Chalmers 1992] D. J. Chalmers: "Subsymbolic Computation and the Chinese Room", In: [Dinsmore 1992], pp. 25–48

[Christiansen | Chater] M. H. Christiansen, N. Chater: "Connectionism, Learning and Meaning" Connection Science 4 (1992), 227-252

[Churchland 1992] P. M. Churchland: "Matter and Consciousness. A Contemporary Introduction to the Philosophy of Mind" Revised Edition, A Bradford Book, MIT Press, Cambridge, MA, 1992

[Clapin 1991] H. Clapin: "Connectionism isn't Magic" Mind and Machines 1 (1991), pp. 167-184

[Clark 1989] A. Clark: "Microcognition: Philosophy, Cognitive Science, and Parallel Distributed Processing", A Bradford Book, MIT Press, Cambrodge, Mass., 1989

[Clark 1992] A. Clark: "The Presence of a Symbol" *Connection Science* 4 (1992), pp. 193–205

[Cummins 1991] R. Cummins: "The Role of Representation in Connectionist Explanations of Cognitive Capacities", In: [Ramsay et al. 1991], pp. 91–114

[Dennett 1978] D. C. Dennett: "Brainstorms. Philosophical Essays on Mind and Psychology", Hassocks, Sussex 1978

[Dennett 1986] D. C. Dennett: "The logical geography of computational approaches: A view from the North Pole. Quoted in: [Dennett 1991b]

[Dennett 1991] D. C. Dennett: "Consciousness explained", Little, Brown & Co, Boston 1991

[Dennett 1991b] D. C. Dennett: "Mother Nature Versus the Walking Encyclopedia" In: [Ramsay et al. 1991], pp. 21–30

[Dinsmore 1992] J. Dinsmore (Ed.): "The Symbolic and Connectionist Paradigms. Closing the Gap", Lawrence Erlbaum, Hillsdale, N.J., 1992

[Dreyfus 1972] H. Dreyfus: "What Computers Can't Do" Harper & Row, New York 1972

[Dreyfus 1992] H. Dreyfus: "What Computer Still Can't Do", M. I. T. Press, Cambridge, MA, 1992

[Fetzer 1991] J. H. Fetzer: "Philosophical Aspects of Program Verification" *Mind and Machines* 1 (1991), pp. 197-216

[Fodor 1981a] J. A. Fodor: "The Mind-Body Problem" *Scientific American* 244 (1981), pp. 124-133

[Fodor 1981b] J. A. Fodor: "Methodological Solipsism Considered as a Research Strategy in Computer Psychology" In: J. Haugland (ed.): "Mind Design", M. I. T. Press, Cambridge, MA, 1981, pp. 307-338

[Fodor 1990] J. A. Fodor: "Why there STILL has to a language of thought",In [Partridge/Wilks 1990], pp. 289-305

[Forrest 1991] S. Forrest (Eds.): "Emergent Computation", M. I. T. Press, Cambridge; MA, 1991

[Frixione/Spinelli 1992] M. Frixione, G. Spinelli: "Connectionism and functionalism:the importance of being a subsymbolist" *JETAI* 4 (1992), pp. 3–17

[Goschke/Koppelberg] T. Goschke, D. Koppelberg: "The Concept of Representation and the Representation of Concepts in Connectionist Models", In: [Ramsay et al. 1991], pp. 129–162

[Harnad 1991a] S. Harnad: "The Symbol Grounding Problem" In: S. Forrest (ed.): "Emergent Computation", A Bradford Book, M. I. T. Press, Cambridge, MA, 1991, pp. 335-346

[Harnad 1991b] S. Harnad: "Other bodies, Other Minds: A Machine Incarnation of an Old Philosophical Problem" *Mind and Machines* 1 (1991), pp. 43-54

[Hastedt 1988] H. Hastedt: "Das Leib–Seele–Problem. Zwischen Naturwissenschaft des Geistes und kultureller Eindimensionalität" Suhrkamp, Frankfurt 1988

[Haugeland 1981] J. Haugeland (Eds.): "Mind Design", A Bradford Book, M. I. T. , Cambridge, MA, 1981

[Haugeland 1991] J. Haugeland: "Representational Genera", In: [Ramsay et al. 1991], pp. 61–89

[Hayes et al. 1992] P. Hayes, S. Harnad, D. Perlis, N. Block: "Virtual Symposium of Virtual Mind" *Mind and Machines* 2 (1992), pp. 217-238

[Heidelberger 1993] M. Heidelberger: "Was erklärt die Informatik?" In: [Schefe et al. 1993], pp. 13-30

[Heidelberger 1993a] M. Heidelberger: "Die Wirklichkeit emergenter Eigenschaften" In: Gesellschaft f. Philosphie in Deutschland (Ed.): "Neue Realitäten. Herausforderung der Philosophie", Berlin 1993

[Horgan/Tiensen 1991] T. Horgan, J. Tiensen: "Connectionism and the Philosophy of Mind", Kluwer, Dordrecht etc. 1991

[Iran-Nejad/Homaifar 1991] A. Iran-Neja, A. Homaifar: "Assoziative und nicht-assoziative Theorien des verteilten Lernens und Erinnerns", In [Schmidt 1991], p. 206-249

[Keil 1993a] G. Keil: "Is the Computational Metaphor of Mind Intentionalistic or Naturalistic?" To appear in: Meggle/Wessels (Eds.): Analyomen. Akten des 1. Kongresses der Gesellschaft für Analytische Philosophie, Berlin/New York 1993

[Keil 1993b] G. Keil: "Kritik des Naturalismus", de Gruyter, Berlin 1993

[Kuhn 19 73] H. Kuhn: "Entstehung des Lebens: Bildung von Molekülgesellschaften. I: Forschung 74", Frankfurt a.M. 1973

[Kutschera 1982] F. von Kutschera: "Grundlagen der Erkenntnistheorie", de Gruyter, Berlin 1982

[Langton 1989] C. G. Langton: "Artificial Life", Addison Wesley, New York 19989

[Lycan 1991]: W. Lycan: "Homuncular Functionalism Meets PDP", In: [Ramsay et al. 1991], pp. 259–287

[Margolis 1980] J. Margolis: "The Trouble with Homunculus Theories." *Philosophy of Science* 47 (198o), pp. 244–259

[Marr 1982] D. Marr: "Vision: A Computational Approach" Freeman & Co, San Francisco 1982

[McClamrock 1991] R. McClamrock: "Marr's Three Levels: A RE-Evaluation" *Mind and Machines* 1 (1991), pp. 185-196

[Newell/Simon 1976] A. Newell, H. A. Simon: "Computer Science as Empirical Inquiry: Symbols and search" *Communications of the ACM* 19 (1976), pp. 113-126

[Partridge | Wilks 1990] D. Partridge, Y. Wilks (Eds.): "The foundations of artificial intelligence", CUP, Cambridge 1990

[Pylyshyn 1984] Z. Pylyshyn: "Computation and Cognition" M. I. T. Press, Cambridge, MA, 1984

[Ramsay et al. 1991] W. Ramsay, St. P. Stich, D. E. Rumelhart (eds.): "Philosophy and Connectionist Theory." Lawrence Erlbaum, Hillsdale, N. J., 1991

[Ramsay/Stich/Garon 1991]: "Connectionism, Eliminativism, and the Future of Folk Psychology", In: [Ramsay et al. 1991], pp. 199–228

[Rosenberg/Sejnowski] C. R. Rosenberg, T. J. Sejnowski: "Parallel Network That Learn to Pronounce English Text" Complex Systems 1 (1987), pp. 145-168

[Roth 1991a] G. Roth: "Neuronale Grundlagen des Lernens und des Gedächtnisses", In: [Schmidt 1991], pp. 127–158

[Roth 1991b] G. Roth: "Die Konstitution von Bedeutung im Gehirn", In: [Schmidt 1991], pp. 360–370

[Rumelhart/McClelland 1986] D. E. Rumelhart, J. L. McClelland, PDP Research Group (eds.): "Parallel distributed Processing", 2 vols., M. I. T. Press, Cambridge, MA, 1986

[Schefe 1987] P. Schefe: "On definitional processes in knowledge reconstruction systems." *IJCAI '87*, pp. 509–511

[Schefe 1991] P. Schefe: "Künstliche Intelligenz. Überblick und Grundlagen", Bibliographisches Institut, Mannheim 1991

[Schefe et al. 1993] P. Schefe, H. Hastedt, Y. Dittrich, G. Keil (Eds.): "Informatik und Philosophie." To appear: Bibliographisches Institut, Mannheim 1993

[Schmidt 1991] S. J. Schmidt (Ed.): "Gedächtnis. Probleme und Perspektiven der interdisziplinären Gedächtnisforschung.", Suhrkamp, Frankfurt 1991 (stw 900)

[Schwarz 1992] G. Schwarz: "Connectionism, Processing, Memory" *Connection Science* 4 (1992), pp. 207–226

[Searle 1980] J. R. Searle: "Minds, Brains and Programs", *The Behavioral and Brain Sciences* 3, pp. 417–424

[Searle 1990a] J. R. Searle: "Is the Brain a Digital Computer?" *Proceedings and Addresses of the American Philosophical Association* 64/3 (1990), pp. 21–37

[Searle 1990b] J. R. Searle: "Consciousness, explanatory inversion, and cognitive science" *Behaviorial and Brain Sciences* 13, pp. 585–642

[Searle 1992] J. R. Searle: "The Rediscovery of the Mind",A Bradford Book, M. I. T. [Partridge/Wilks 1990] , Press, Cambridge, Mass. 1992

[Smolensky 1990] P. Smolensky: "Connectionism and the Foundations of AI", In: [Partridge/Wilks 1990], pp. 306–326

[Stegmüller 1979] W. Stegmüller: "Hauptströmungen der Gegenwartsphilosophie, Band II", Kröner, Stuttgart 1979

[Tetens 1993] H. Tetens: "Informatik und die Philosophie des Geistes." To appear in: [Schefe et al. 1993]

[Tugendhat/Wolf] E. Tugenghat, U. Wolf: "Logisch-semantische Propädeutik", Reclam, Stuttgart 1983

[Turing 1950] A. Turing: "Computing Machinery and Intelligence" *Mind* 59 (1950), pp. 433-466

[van Gelder 1992] T. van Gelder: "Defining 'Distributed Representation'" *Connection Science* 4 (1992), pp. 175-190

Ackowledgement

I would like to thank The Institute for Cognitive Studies of the University of California at Berkeley for inviting me as a Visiting Scholar, especially Hubert Dreyfus, for his support and some suggestions for the improvement of an earlier version of this paper. I am also obliged to Heiner Hastedt, Rüdiger Menzel, Matthias Jantzen, Klaus von der Heide who made suggestions for further improvement. I thank John Searle and Geert Keil in particular for discussions of the main issues of the paper, and M. Simmons for a careful reading resulting in suggestions for linguistic improvement.