THE UNIVERSITY OF CALGARY

Computation and Causality

by

Peter Hajnal

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF ARTS

DEPARTMENT OF PHILOSOPHY

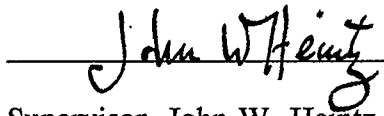
CALGARY, ALBERTA

DECEMBER, 1994

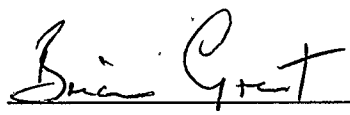
THE UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

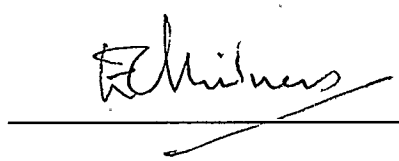
The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "Computation and Causality" submitted by Peter Hajnal in partial fulfillment of the requirements for the degree of Master of Arts.



Supervisor, John W. Heintz, Department of Philosophy



Brian E. Grant, Department of Philosophy



Eric C. Milner, Department of Mathematics and Statistics

December 21, 1994

Date

Abstract

The thesis examines various arguments proposed to define the methodological role computations can or should play in the philosophy of mind and cognitive science. A distinction is made between assumptions of AI and computational theories of *mind* (CTM). It is argued that attempts at defining exactly the scope and limits of CTM are inconclusive, and their power to persuade does not exceed that of some arguments of Leibniz dating from 1714. Some putative reasons are offered for explaining the persistence and the nature of the controversy surrounding CTM. Based on evidence drawn from the mathematical theory of computations and independent philosophical arguments, it is argued that the notion of a computational physical system is a vague notion, and algorithms appear to play an explanatory role of a magnitude that is unwarranted by their nature.

Table of Contents

List of Figures

CHAPTER I.

Introduction: Leibniz's Mill

1. What Leibniz said

A long time ago, before various 18th, 19th and 20th century revolutions were effected - scientific, political, industrial and otherwise - the German philosopher Gottfried Wilhelm Leibniz came up with an argument to show that there will always be some things which humans will be able to do, but machines will not. This is what Leibniz observed in the Monadology:

"It must be confessed, moreover, that *perception* and that which depends on it *are inexplicable by mechanical causes*, that is, by figures and motions. And, supposing that there were a machine so constructed as to think, feel and have perception, we could conceive of it as enlarged and yet preserving the same proportions, so that we might enter it as into a mill. And this granted, we should only find on visiting it, pieces which push one against another, but never anything by which to explain a perception. This must be sought for therefore in the simple substance and not in the composite or in the machine." (Leibniz (1951) p. 536.)

A further remark taken from the same work illuminates the point further:

"Thus each organic body of a living being is a kind of divine machine or natural automaton, which infinitely surpasses all artificial automata. Because a machine which is made by man's art is not a machine in each one of its parts; for example, the teeth of a brass wheel have parts or fragments which to us are no longer artificial and have nothing in themselves to show the special use to which the wheel was intended in the machine. But nature's machines, that is, living bodies, are machines even in their smallest parts *ad infinitum*. Herein lies the difference between nature and art, that is

between the divine art and ours." (Leibniz (1951) p. 546.)

These remarks date from 1714. What is remarkable about them - hence my remarks at the beginning - is that we should know better. But we don't. This essay is about how these Leibnizian problems are still with us, and what is even more disturbing, some contemporary arguments are very similar to these despite the fact that they date from nearly 300 years ago. Some philosophical problems of current interest are even older of course, but one feels that these are typically problems about which technological and conceptual advances should have been able to say something definitive by now. It seems that there is something about the core of the problems which prevents this.

2. Two problems instead of one

In some sense, of course, we do know better. Even schoolchildren will be able to point out some "mistakes" in the above. Faced with the first quote, the kids will tell us that Leibniz would have a hard time walking into any electronic device to figure out what is pushing against what, and he would have an even harder time walking around inside the magnetic grooves of a videotape. The kind of devices Leibniz was familiar with were built on mechanical principles, today we work with electrical and chemical ones as well.

So far so good. The second quote, however, is still to the point. We don't have a machine which thinks and many people believe that we never will, because it is in principle impossible to build one artifically. Whoever thinks this is in much the same position as Leibniz; he will have to come up with an argument like the one in the second quote about the *difference* between "us" and machines which makes this impossible. Leibniz's argument seems to be no good. Once again, the kids will tell us that even organic stuff is made out of inorganic stuff - it's all atoms, whether it be cogwheels or brains.

Despite all this, I think that much of today's theorizing about a certain problem still amounts to no more than - metaphorically speaking - walking around inside Leibniz's mill. This problem concerns the validity of a certain set of assumptions whose heuristic

value is accepted to various degrees within a number of disciplines. Whether these assumption have a common core is a question in itself (which I shall be trying to answer). Assuming that they do (although it be blurred at the edges) I shall group them under the label of the "Computational Theory of Mind". What the people walking around in the mill are trying to find out, is whether the computational theory of mind (or CTM for short) should be dismissed or embraced and for what kinds of reasons.

There is a subtle difference between arguments advanced for and against CTM. Those who argue *for* the theory are in some sense much closer to the spirit of Leibniz's reasoning than those who disapprove of it. This is because arguments in support of the claims (whatever they are) - sometimes at least - involve arguments about how there is no alternative to CTM if one is going to construct an explanation of mental life, which is to be consistent with the basic assumptions and high level of sophistication of contemporary scientific theories. Construals like this - similarly to Leibniz - assume that we are faced with a choice between two alternatives. Sharing the commitment to the insight inherited by thinkers of the 18th century from the previous one, we continue to believe, that whatever metaphysical assumptions we *have* to make must necessarily be constrained by our scientific theories. Leibniz - once having persuaded himself that science lacks the power to explain certain phenomena - had a much greater degree of freedom than us (for reasons implicit in the children's reasoning) in making metaphysical assumptions. Some who argue that the hypotheses of CTM are indispensable, do so by claiming that otherwise the strategy fails: there is nothing to substitue for Leibniz's second argument which could save us from dualism, mysticism or religious teleology. Leibniz was limited to "mechanical" explanations. Today, in some sense, we have a full palette. We paint with one of the colours or we don't paint at all.

This is one type of argument, and perhaps it is accidental that it occurs more often in writings in agreement with CTM. Those who argue against it - unless explicitly criticizing believers arguments constructed along these lines - usually come up with stories of the kind in the first quote. These are the people who think that the situation

described in the Monadology is precisely the kind CTM puts us into, hence to be rejected for reasons that are much like Leibniz's own. Chapter one of my essay will look at arguments of philosophers of like persuasion together with how some positive construals attempt avoiding the objections. The second chapter is devoted to an argument of the sort outlined above.

I should mention a common feature of the arguments I examine in the following chapters. They both begin by *redefining the concept of computation.* In other words, it's almost as if there was disagreement with respect to the very concepts in terms of which the theory is framed. And indeed, this state of affairs was one of the reasons why I myself became interested in the problem. Having studied the mathematics of computation it seemed to me that I had understood it, and I was puzzled and intrigued about both the heatedness and the stubbornness of the debate. The heatedness seemed mysterious because it doesn't strike me as obvious *what it is about the concept of computation which allows for such wildly contradictory opinions.* The stubbornness is the other side of the same coin. It would appear to the naked eye as if there hadn't been much shifting of emphases or topics within, or change in the character of the debate since the early seventies. Classic works such as Hubert Dreyfus's "The Limits of Artificial Intelligence" or David Marr's writings, seem to be raising the same issues one encounters in the discussions of today. I believe, however, that if nothing else, the edges have been sharpened. It is much clearer today what computers <u>can</u> do even if what they cannot is still mysterious. In other words the prospects and limits of "Artificial Intelligence" and its difference from CTM[1] is much clearer now then it was when Dreyfus wrote his

---

[1] There are of course some classical papers which even "back then" dealt strictly with the problem of CTM. Putnam's papers on the topic for instance argue, that the mind-body problem arises as the result of a certain "deviant" question where it this same kind of deviance which arises in connection with demanding a Turing-machine to report it's own condition. It is quite impressive that Putnam's papers haven't aged at all. I shall not be concerned with his arguments in this essay, but I warmly recommend them to anyone interested in the topic. See: Putnam (1975).

book[2], or when the first computerized natural-language translation project was conceived. Nonetheless, something still seems to be missing. The arguments I treat in the next two chapters are more recent ones - common sense tells me they shouldn't be. I shall attempt to question common sense about this and try to find out how "it" came to frame this opinion.

3. Something about CTM

Of course, one thing I have assumed so far is that CTM is the modern equivalent of Leibniz's problem. This fact (partly at least) is one of the things which should gradually become apparent as we progress toward the end of the essay. However, there are a few things about the difference between artificial intelligence (AI) and CTM which I should mention before actually moving into the arguments themselves.

Whereas CTM is a theory about the nature of the mind, or if you like: the nature of our thought processes *as a whole*, AI is not actually committed to any assumptions of this sort. There could be a misunderstanding here for there is something about computers which justifies the validity of the AI projects; and what justifies the currency of CTM is also - partly - something about computers. But the two things are not the same. It is more or less a common assumption nowadays about the methodology of psychology that it cannot do without mental entities, and CTM is an attempt at explaining the nature of these mental entities. This state of affairs is expressive of the fact that what has been formerly an argument for AI's actually being what the name implies, i.e. an attempt to

---

[2] Some of Dreyfus's criticisms imply for instance that chess programs will never outdo human players. (See: Dreyfus (1972).) He also ridicules enthusiasm about the solution of many other tasks by artificial means already appearing on the horizon. Although he was right about natural-language translation, today's chess-programs play at the grand-master level. Whatever one may think about this, one thing is for sure: those arguments, at least, are sure to be faulty. I touch on the issue of whether this tells us anything about the nature of "intelligence" in section 3. of this introduction. It is a separate topic that our understanding of the mathematics of computability has also advanced enormously over the years. This is true especially with respect to central questions having been crystallized and the difficulty of their solution becoming apparent.

create "artificial intelligence", this is no longer true. This change is reflected in an amusing anecdote recounted by H.G. Pagels.

"A few years ago, I asked a colleague of mine at Harvard what was the future of Artificial Intelligence. He simply said: 'Heinz, if you took the smartest two dozen people of the eleventh century and put them in a room together and instructed them to put together a model of the physical universe, there's no question that they would come up with something that would be absolutely brilliant. But it would be all wrong, because the concepts were not to be invented until several centuries in the future. That's similar to the case for the AI proponents - they are very smart people, but the right concepts are not yet available.'

"People who work in artificial intelligence research have to contend with critics of that kind, and I'm sure we'll hear a good deal about that. I once asked Marvin Minsky why this field of study was ever called artificial intelligence. I said, 'Why didn't you call it something more general, like cognitive science?' And Marvin responded: 'If we ever called it anything other than artificial intelligence, we wouldn't have gotten into the universities. Now that we're in, and the philosophers and the psychologists know that we're the enemy, it's too late.'" (In: Pagels (1984) p. 138.).

What clearly appears from this short discussion is that the ambitions of artificial intelligence are no longer what they once were, that is quae practical enterprise, its aspirations are at most the realization of some intelligent processes by mechanical means, but not the realization of the process of intelligence as a whole. What originally fueled the AI project was the fact that a computer had some remarkable phenomenological properties, namely it was the first machine which was actually capable of behaving - if only in superficial ways - intelligently. On the other hand, there is some evidence drawn from mathematical sources that the computer is in a particular sense the best machine we will ever have, and hence if there is some practical way to disprove Leibniz's thought, the computer will have to feature in the demonstration. The connection betwen these two ideas is neatly manifest in the so called Turing-test. As Turing-machines will feature in

this paper, I shall also give a superficial description of Turing-machines.

The Turing-test is the classical "behaviourist" test for intelligence. It is ingenious in two senses: first, given the disagreements over the definition of intelligence, it exploits the fact that there is at least one kind of agent, namely the human agent which can certainly be said to possess it[3] and second, it does not assume anything about the internal mechanisms of the machine which is to be tested for intelligence (precisely the domain of CTM).

The scenario involves a machine and two humans. They are all placed in different rooms which are connected by computer terminals through which they can type messages to each other. One of the humans, the interrogator, is given the task of finding out which of the other two is the human and which is the machine. The way he is supposed to do this is by typing questions on his terminal and sending them to the room of his choice. If it turns out to be impossible to distinguish on the basis of the answers he receives between the human and the machine, then the machine is said to have passed the Turing-test for intelligence.

The Turing-test is called thus because it was suggested by the mathematician-philosopher Alan Turing who is also known for inventing Turing-machines - a mathematical formalization of effectively computable functions.

A Turing-machine consists of two parts: a two way infinite tape divided into identical cells and a read/write head. The reader should imagine the tape as literally being a tape such as an infinite roll of toilet paper which is marked into identical squares and the head as being a box with a pencil sticking out of it which can make marks on tape. As a next step we imagine the box to be relatively "smart" in the sense that it can "recognize" certain "symbols" written on the tape and it can "respond" by "doing" something. This

---

[3] I think the performance-competence distinction is relevant here. We are testing for the *possession* of an IQ not for the numerical value although this might be a conceptual impossibility (imagine something with an IQ of 20 ;would that still count as intelligence?). See Chomsky (1980) p. 201-5 for a discussion.

"doing" can consist of: nothing, erasing the symbol and writing something else in place of it and/or moving away from the cell in either direction. Furthermore, what the machine will do after having read a symbol from a certain cell shall depend on nothing else except the symbol itself and the current "state" of the box. (I am inclined to use the term "machine" as denoting the box only but in fact the machine is the box together with the tape plus the symbols it can recognize. It should be clear from the context what I mean.) The limitation put on the symbols and the states is that there can only be a finite number of both. We can also suppose that there is a special beginning state of the machine (the one which it is in before having read anything) and a special end-state (after having reached this state it stops).

The Turing machine thus "computes" on strings of symbols. The input is a sequence of symbols which someone (or another Turing machine) writes on the tape. When the machine is turned on, it will fumble around for a while going from cell to cell, now and then erasing and writing symbols, and if we are lucky, then it will stop eventually (although there is nothing to guarantee that it will). The output is the string that sits on the tape after this session. (For a serious introduction to Turing-machines see Hopcroft and Ullman (1969) or any other introductory textbook to automata theory. For an informal introduction see Hofstadter (1979) or Penrose (1989))

Now, even knowing this much about Turing-machines, one thing is obvious, that is that *the Turing-machine is the kind of thing which is at least capable of participating in the Turing-test*, that is, it can take sentences in English as input and give sentences of English as output. As a matter of fact, at first glance it might not be so obvious that something as abstract and simple as a Turing machine can do this, but it is true. And in a sense no actual computer has more computing power than a Turing-machine, hence the model is about as general as it can get.

These and other reasons of this sort (e.g., that the computer can be said to "perceive" things in the environment and react to them, i.e., be an instantiation of S/R psychology) were what fuelled the original AI enthusiasm and also the bitter attacks against it. The

division between CTM and AI is essentially due to the realization that behaviourism as a theory of psychology is insufficient, hence the Turing-test does not say anything about mentality, and what results in AI have to say about certain problems is, perhaps, that they did not need much intelligence in the first place. I have discussed these things in the introduction, because it is important to keep in mind, that for the arguments I am going to deal with, it is not the phenomenology of the computers that is important but - in a sense - their ontology. The methodological assumption, though having been separated from the practical one, is still there. A cursory glance at Turing-machines and what their existence does and does not imply in terms of theory was necessary before moving into the actual controversy. The rest of what must be said about machines will emerge along the way. In the last chapter I shall return to Turing-machines and say a few more things that are relevant in the light of intermediate developments. For now, this should suffice.

CHAPTER II.

Against computationalism

In this chapter I will look at some variants of arguments that are devised to eliminate CTM once and for all from any serious discussion about the mind. Although this is intended as a survey section, I have chosen to organize it around a paper dating from 1985 by D.H. Mellor[4], which the author presented at the Australasian Cognitive Science Symposium held at the University of New South Wales in Sidney. Ironically, Jerry Fodor who attended the same meeting read a paper entitled "Why there still has to be a Language of Thought"[5], which is a short overview of what is perhaps the most influential formulation of CTM and a confirmation of the fact that in spite of the opposition the theory is alive and kicking. Mellor's article will offer a glimpse of the typical style of argumentation, and the nature of his presentation is such that it will allow me to introduce some other arguments along the way. Thus this chapter is a criticism of the paper in some sense. I do not think that the conclusions in general - that not much of the mind is in fact a computer - follow from the arguments. On the other hand, and this is perhaps more important, there are many intriguing suggestions about how to approach the problem and due partly to the polemical nature of the arguments, the paper covers a very wide range of topics.

1. The question and how to answer it

Section 1 begins by stating the question: "How much of the Mind is a computer?" (Mellor p. 47.) The answer will be: not much. The argument proceeds in what are

---

[4] D.H. Mellor: "How Much of the Mind is a Computer?" in: Slezak and Albury (1989).

[5] Jerry Fodor: "Why there still has to be a Language of Thought" in: Slezak and Albury (1989). Also in: Fodor (1987).

approximately two stages. First, Mellor gives a definition of "computation" and then he tries to show that there are many mental processes which can not be conceived of as computational in this sense. It is evident that whatever definition we adopt for a process being computational, it must be "wide" enough to encompass "actual" computations such as those implemented on a "general purpose electronic digital computer" and also, if the argument is going to work, it shouldn't be so generous as to admit interpreting anything mental automatically as computational. Mellor even seems to give some headway to the computationalist in the sense that, instead of beginning with some mathematical formalization of effective computability, he proposes a definition which is intended to encompass all processes plausibly viewed as computational (with Turing-computations included as a subset). We are reminded of one of Fodor's methodological principles that support his adoption of the computational theory of mind: "A remotely plausible theory is better than no theory at all". (Fodor (1975)) Mellor would like to show that computationalism is not even a remotely plausible theory because, given even the most liberal definition of computation, most mental processes do not qualify for being in its extension. The first part of the paper (sections 1-4) is thus concerned with forming a new theory of computations and to:

"...shake the conviction that we know what computers are because we are familiar with the general purpose programmable electronic digital computers that are their modern paradigms. They are what give our question its interest and its real sense: 'How much of the mind is to be explained in terms of such machines?' And this sense seems clear because we have a largely agreed vocabulary for describing the working and uses of these machines: such terms as 'computation', 'representation', 'information', 'data', 'processing', 'program', 'syntax', 'semantics' and 'algorithm'. This all generates a spurious consensus about what computers are - spurious because different people read key terms in this list, notably 'information' and 'representation' very differently." (Mellor p.47.)

We are again reminded of Fodor, who when arguing against Wittgenstein's position on

private language, states: "...there *are* such things as computers, and whatever is actual is possible." (Fodor (1975) p. 86.) In particular we also know in quite a clear sense what a computer is capable of and also what a computation is, namely in the mathematical sense. Notably the terms 'information' and 'representation' are in no way needed for a correct definition of effective computability. Thus it seems that Mellor is talking about the metaphorical usage of the terms and what we seem to be looking for are new consistency criteria for describing certain processes metaphorically in computational terms, but then again I may be wrong. What I would like to point out is that when the author states on page 48. that "The only assumptions I make here are those I shall make explicitly", it does not follow readily that the explicit assumptions themselves are more valid for being explicit as opposed to latent. In any case I will try to take the author's insistence on explicitness (and non-ambiguity) quite literally, although one of the morals of this section will be that it is almost impossible to adhere to a condition this strict. Where the concepts and the problem are fuzzy, arguments naturally tend to be approximate.

2. A definition of computation

The search for the correct definition of computation begins in section 2. of Mellor's paper and I shall present the arguments in some detail. It seems to me that this is where it is "all" decided.

"First, all parties agree that a computer is defined by what it does, namely compute. It doesn't matter how it computes. When, for instance, one computation needs the result of others, it is immaterial whether the others are done in series (one after the other) or in parallel (together). Nor does it matter whether a computer computes with silicon chips or with brain cells. It need not even compute with matter; a spiritual computer is not a contradiction in terms. Whether mental processes are computations is independent of whether they are material. (So fortunately we can set that traditionally vexed question aside.)" (Mellor p. 48.)

Personally, I have a strong conviction that there is no such thing as an immaterial process, or at least it is not straightforward fact that there is. It is perhaps not even so important (especially in the light of later developments) to spend much effort on interpreting this paragraph. However - given my (purely ideological) qualms about immaterial processes - the interpretation I would propose is, that a series of changes (in whatever medium the changes take place) will count as a candidate for a computation only by virtue of the nature of relations holding between the beginning state, the endstate and the intermediate states. What does not count is how these changes were effected. Thus even if they were effected by magic but they are the right kind of states with the right kind of relations holding between them, the process will count as a computation. For instance (although at this stage in the argument we have no evidence for our intuition) an input and an output state without intermediate states will rarely count as a computation although in special cases (barring further specifications) it could, namely the identity function would seem to count as a computation function in any case. Under this interpretation there is an important limitation lifted from the definition, namely that we will not have to take into account any theoretical limitations placed on the symbol manipulating capacities of physical devices. In at least one important sense this characterization of "computation" in Mellor's sense (or what we have of it so far) is analogous to computability as defined by Turing machines. In the latter case it is irrelevant how the transition between states is effected in the TM. In yet another formulation: a putative computation is a series of changes taking place in some medium - effected by an agent. In deciding whether the process is a computation, the nature of the agent is irrelevant. In fact the presence of the agent is not even necessary if we view the process globally. (A Turing-machine can be viewed as a device which effects changes in strings of symbols, but a particular computation of a Turing-machine can be viewed as simply a sequence of changes in the physical system consisting of the machine *taken together with* the string on the tape.)

As it will turn out, however, these are minor (though interesting) considerations. The features that will figure in the foreground of Mellor's arguments about computationality begin emerging in what follows:

"Next, however computing is done, it is agreed to be the processing of information, as opposed to the processing of matter. A food processor making pate from its ingredients is processing matter: a word processor making a description of the pate from a description of its ingredients is processing information. What is the difference?

"The main difference is that information is true or false, whereas matter just is. Pieces of information are what I shall call "propositions", whether they are expressed or embodied in sentences, pictures, computers, beliefs or any other way. A proposition, e.g., *the earth is round*, corresponds to a state of affairs (the earth's being round), which may or may not obtain. If it does (if the earth *is* round), it is a fact, which makes the proposition true; if not, the proposition is false." (Mellor p.48.)

These two passages contain a multitude of suggestions. The first paragraph suggests that as a minimal requirement on the sequence of states that we take to be a computation the input and the output have to be pieces of information. This sounds problematic in the light of our analysis above which led us to conclude that the input and the output have to be physical. In this sense it is matter of some sort that is being processed during the computation. What we have established above is that the "device" that does the processing, that is - induces changes in the configuration of the particular matter - may be, as it were, angelic in nature. What we have to be careful about at this point is not to take it to be implied in Mellor's suggestions that whatever material sequence is involved in the computation it must have an informational "interpretation", that is it must constitute a sequence of "representations" of propositions. This assumption at least has not been made *explicitly* as yet, and I would like to be very careful in keeping my criticisms (or approvals) gradual. We mustn't forget that we are engaged in a project the aim of which is to build up a definition of computation and computational concepts from scratch.

The second paragraph interprets "piece of information" as a proposition (that is a "sentence type" as opposed to a "sentence token") which is a rather idiosyncratic (although not unheard of) concept of information. This may not be very important however. Since we are looking for a definition of computation we have a certain free

hand in choosing our primitives, constraints are important when we employ our construct for classificatory purposes. What *is* significant is the explicit emphasis on a computation being the processing of something non-material, for even if we were tempted to apply the familiar semantic - syntax distinction after the first paragraph, what follows makes it clear that it would not be an unproblematic interpretation and later developments will confirm this. As we will see, figuring out the nature of "syntax" is one of Mellor's central concerns. We must follow the road that leads up to it however, and, I think it is correct to say that after these two paragraphs we are minimally left with a dilemma as to the nature of a "computation". Everything, I assume, is either material or nonmaterial. I will leave it for later to decide which if any of the two is preferred by the author.

In what follows in the text the author makes some comments on the nature of propositions. First he allies himself explicitly with the so called "intentional realists", those philosophers of mind who claim that propositional attitudes have *ontological reality*. In this he sides with Fodor and others (Fodor is the only one who is explicitly mentioned) who claim that there is no conceivable explanatory framework of the mind which can do without such primitives as beliefs, desires, hopes, fears, etc. Mellor also makes explicit the distinction (implicit in the above) that propositions as such do not exist materially (at least not outside the brain), and hence a computation processes not the propositions themselves, but tokens of the propositions. This is a very important point so I shall quote the author's own formulation:

"The earth must be somewhere, and facts about it will have causes and effects. But propositions about it, whatever they are [Pieces of information allegedly. My note. P.H.], are nowhere in particular, and neither affect nor are affected by anything. Yet computers process propositions causally, in definite places and definite times. They must therefore process them indirectly, by processing tokens of them that do have causes, effects and spatio-temporal location. A proposition therefore, is a *type*: again unlike a piece of matter, the very same piece of information may be processed many times, and in any number of places at once. The information output by a single computation will therefore not be a new proposition, but a new token of a proposition, generated by a

causal process from input tokens of other propositions." (Mellor p.49.)

I think that it is important to notice that the sense in which computation is taken here contradicts - if only in minor ways perhaps - the suggestions we have been dealing with above. First of all, computation is explicitly conceived of here as a *causal* process whereas in the foregoing we were quite content with settling for the possibly mysterious way whereby the sequence of change in matter is effected. It also contradicts our second observation: that computation is the processing of something nonmaterial, namely (and not very importantly) propositions. In the last sentence of the above paragraph what is explicit is, that the information which will be the output of the computation will, in fact, be the token of a piece of information, that is a (token) proposition. One gets a sense of the author being keen for some reason on refraining from the use of the terms "interpretation" and "representation" which would be very natural in the context. What we can hope for is that they may receive some special and interesting role in the rest of the argument which will also serve to dispel the present ambiguity.

The next remark concerns another important point, namely that "this causal processing of information" should be deterministic in the sense that the output information should be a *function* of the input information in the *mathematical sense*. This means, that given a particular computation, and given any piece of information that can in principle be an input to that computation, there has to be one and only one piece of information designated as output. That is, the computation should compute a function in the mathematical sense with a well defined domain and range.[6]

---

[6] The author's comments run as follows: "The information output by it should be fixed by the information input. That is, it should be a *function* of it in the sense in which (e.g.) birthdays are a function of people but not *vice versa*: people - the birthday function's 'arguments' - have only one birthday each - its 'values' for those arguments; whereas many people share the same birthday. This sense of 'function', since it figures largely in what follows, I must say at once has almost no connection with the concept of a function in biology, anthropology or the philosophy of mind." (Mellor p. 49.) Needless to say, what is being made such a big fuss about here, is the notion of function we all learned in school.

Thus, so far we are not yet certain what the proposed definition of computation amounts to, but we do know that for every computation there corresponds a unique information function (which from now on I will denote by F_i) whose range and domain correspond to the inputs and outputs of the computation. The author's next comment still leaves us a bit in the dark. He says that "..a computation's information function cannot be given just by its causal processing." Before going on to what follows we may remark that according to this, a computation is not identical with either the "information function" or the causal processing. We have here three different entities standing in a certain relation to each other. Without going into details I am going to interpret the above remark as stating the (not so trivial) fact that no physical-symbol has by virtue of its physical properties and by them only a unique semantic interpretation. A simple proof of this fact would be that we can imagine using any symbol to substitute for any letter of the English alphabet.

What the causal processing "gives" is another function (in the mathematical sense again), that is a function from input tokens to output tokens. The author puts this somewhat differently:

"But a computation's "information function" cannot be given just by its causal processing. Causal processes work only on intrinsic properties of the tokens involved. The properties may be chemical, or electrical, or even mental - e.g. being some kind of pain or visual sensation. The range of intrinsic properties is disputed: whether for instance it includes relational properties, like being hotter than something. But no one thinks it includes being a token of anything like a proposition, say that the earth is round. Causal processing can produce a token with the intrinsic shape of the sentence '*The earth is round*', but not one that intrinsically corresponds to that state of affairs." (Mellor p 50.)

The difference that I was referring to above may be clarified in the following way. Denoting the *causal function* by F_c the simplest thing would be to make F_c a function of one variable which takes token propositions to token propositions. However given the

author's insistence on causal processes working on "intrinsic properties" of tokens (and his insistence on the nature of the relations being mathematical) and pairing it with the remarks that follow in the next passage we might have to commit ourselves to something more complicated.

"What causal processing supplies is a function from intrinsic properties of input tokens to intrinsic properties of output tokens. For this causal function to yield an information function, those properties must be correlated with propositions. The correlation need not be one-to-one: tokens with different intrinsic properties - e.g. upper-case instead of lower-case tokens of 'THE EARTH IS ROUND' - may well be processed in the same way. So the relevant intrinsic properties of tokens need not be a function of the correlated propositions. But the propositions must be a function - a 'semantic function' - of them if the causal function is to embody an information function. A semantic function, in short, is what makes a causal process a computation. Any computer, i.e. any causal system for processing information, must impose or exploit some suitable semantic function." (Mellor p. 50.)

Thus here we finally get a definition of a computer as being "...a causal system for processing information" which flatly contradicts the assumption that a computer might be immaterial. Also it is not clear how a "computer" is supposed to "exploit" or "impose" a semantic function. Jumping ahead of ourselves just a little, we may remark that if a computer could do this to itself, our whole problem would be solved (I shall elaborate on this below). It appears simply that Mellor is making an attempt to analyze carefully the simple observation which Fodor puts as "..no computation without representation" and the result is some degree of confusion.

There is one more notion left to introduce which Mellor considers essential for his arguments: *syntax*. Indeed, as I had already mentioned, Mellor considers the concept very important. Unfortunately, the way it is introduced tends to make it difficult for me to see clearly, that in what - according to Mellor - the difference should consist in between one's pretheoretical intuitions, and the carefully analyzed version. The text is

quite simply not very clear. Before I take a shot at interpretation let me introduce an illustrating example (which I will refer to again).

Consider the following "computing system". We have a set of input propositions and a set of output propositions with an information function which assigns outputs to inputs. We assign, say, marble balls of different size to each input proposition and we construct a "box" - the machine - which will have a number of holes of different sizes inside, distributed in such a way that if we drop a ball in at the top it will end up inside the box in a well defined "pot" and where it ends up depends only on the size of the marble (i.e. its diameter, or any other parameter sufficient both for individuation *and* for physical implementation). We also assign some tokens to the output propositions and we set the whole thing up so that when an input ball arrives at its destination it knocks out a token of the proposition designated by the information function. (Notice that the tokens assigned to the outputs do not have to be systematic in any way as opposed to the input tokens.) See Figure 1.

There are some things to notice about this scenario. First of all it seems at least dubious to me that people would like to call what is happening here "computation" although basically we have satisfied all the conditions so far given by the author. We have an information function F_i that pairs propositions, a causal function F_c acting on token propositions and a semantic function F_s assigning semantic values to the tokens and even exploiting some "intrinsic physical property" of the tokens - namely their size.

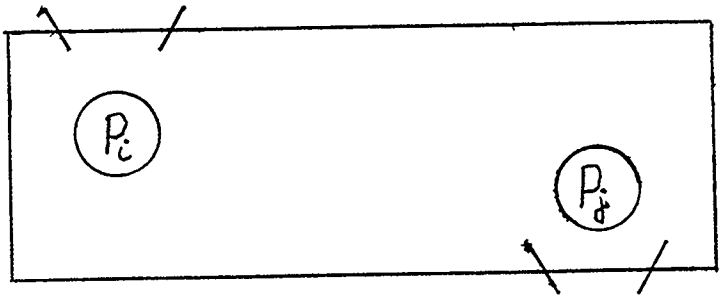


FIGURE 1.

The Ball-machine

We even have representation - although at this point this did not feature in the arguments, and in this sense this "machine" of ours is computational in the sense that physical objects acting in accordance with the laws of Newtonian mechanics are not. (One of Mellor's examples.)

What is missing? Consider this:

"Representing a state of affairs by intrinsic properties of a token proposition, by which it is causally processed, is the core of computing. But computers represent more than states of affairs. They also represent their constituents: e.g. the earth (or the concept of it), which is a constituent of real or supposed facts about it. They do this by using semantic functions that are compositional, i.e. which represent a state of affairs by a token's structural properties: properties that are spatial or temporal functions of properties of its parts, which in turn represent constituents of the state of affairs. Thus the shape of a printed token of '*the earth is round*' is a spatial function of the shapes of the words in it that represent the earth (or the concept of it) and the property or concept of being round; just as the sound of a spoken token of that sentence is a temporal function of the sound of its spoken words. These further functions we may call 'syntactic', because collectively they constitute a computing system's grammatical rules of composition, its *syntax*; just as the arguments of those functions - e.g. English words - are its syntactic elements, its vocabulary." (Mellor p.50.)

And later:

"So in particular a causal function that is a computation is automatically a syntactic function, from the causally relevant types of the input tokens to those of the output token. And the total correlation of a system's syntactic types (the arguments and values of all its syntactic functions) with what those types represent (states of affairs and their constituents) is the system's semantics." (Mellor p. 51.)

I leave it to those who have the time and inclination to attempt an interpretation of just

what "... the total correlation of a system's syntactic types with what those types represent" might be trying to assume "explicitly". I would just like to point out in passing that we get a brand new definition of computation at the beginning of the first paragraph of the quote which is rather different from the previous ones. Nonetheless, we can summarize the present status of computing in the following figure:

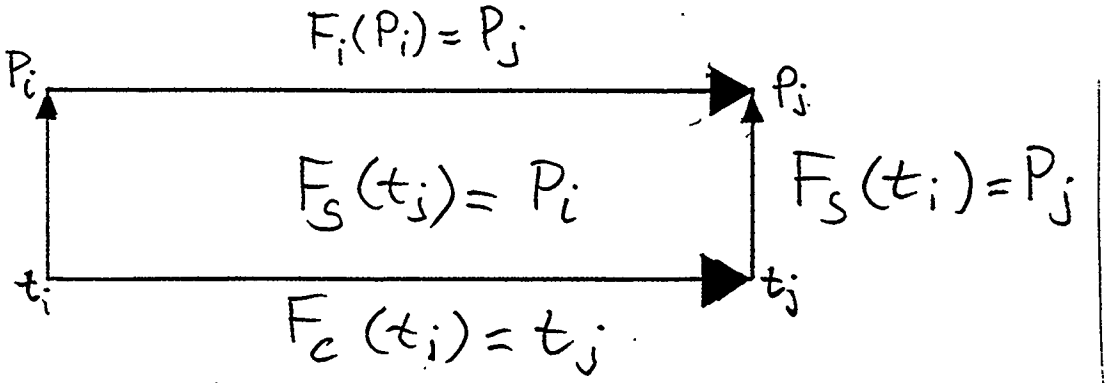


FIGURE 2.
Mellor's computation

Taking everything together it seems that Mellor wants much more than the simple picture of assigning token representations to token propositions. In particular he assumes that our representational system will be "structurally sensitive" to certain semantic relations. This is a reasonable requirement if for no other reason than that, even if someone is familiar with the "ball-machine" described above it is by no means obvious from this alone how to construct a machine which will be able to manipulate tokens in complex ways depending on precisely those physical properties of the tokens which "represent", and also in ways that yield "meaningful" results. So even if in some sense the balls of the "ball-machine" can be said to have syntax it would be difficult to argue that those syntactic properties reflect relations between semantic values of the tokens. This is a difficult issue, but what does seem clear is that there is an issue, namely that it is by no means a simple task to provide a general notion of syntax (limited notions are readily available of course, e.g., the syntax of a formal system is a pragmatic notion.)

I hope that the reader joins me in feeling that there is no such thing as a definition of syntax to be found in the above quotes (at least not one which is independent from

semantics), but that there is certainly something to be said for the notion of syntactic functions indicating the presence of real computations as opposed to what goes on in a simple sorting machine as in the example of Figure 1. I shall soon return to the issue of syntax in more detail.

After this rather tedious overview of section 2 of Mellor's article we are ready to tackle the arguments concerning computation laid out in the main body of the text. That is, we will be ready after getting over the initial shock occasioned by the first sentence of section 3 which sums up what has gone before: "So far, I hope, so trite."

3. What is computational?

The author continues:

"But what does it take to process information, i.e. to turn a causal process into a computation? That will determine what a computer is, and hence the sense of our question. " (Mellor p. 51.)

The rhetorical question takes us by surprise. In the preceding we have in fact received quite a thorough analysis of what computing is - in fact we have even been faced with some mutually incompatible definitions. It seems for instance that the answer to the question has been already given. What turns a causal process into computation is the processing of propositions, and we even got an account of how that is supposed to happen (i.e., by exploiting some complex syntax etc.) The point Mellor is getting at is made on page 55. "We must take computers to embody the information they process in an actual and not merely in a dispositional sense." This point is illustrated by an elaborate example which I will not attempt to reproduce here. Briefly, "veridical perception" is compared to a situation where two forces act on some object resulting in the acceleration of the object. Mellor shows how both situations can be interpreted as processing propositions, but whereas the actual processing of tokens embodying information is at least a plausible explanatory model for what happens in the first case,

the accelerating body plainly does not process anything like physical tokens of propositions expressing the magnitude and direction of forces acting on it. (I reproduce the figures here for the interested reader to ponder. They are sufficiently detailed as to be able to convey information without my discussing them in depth.)

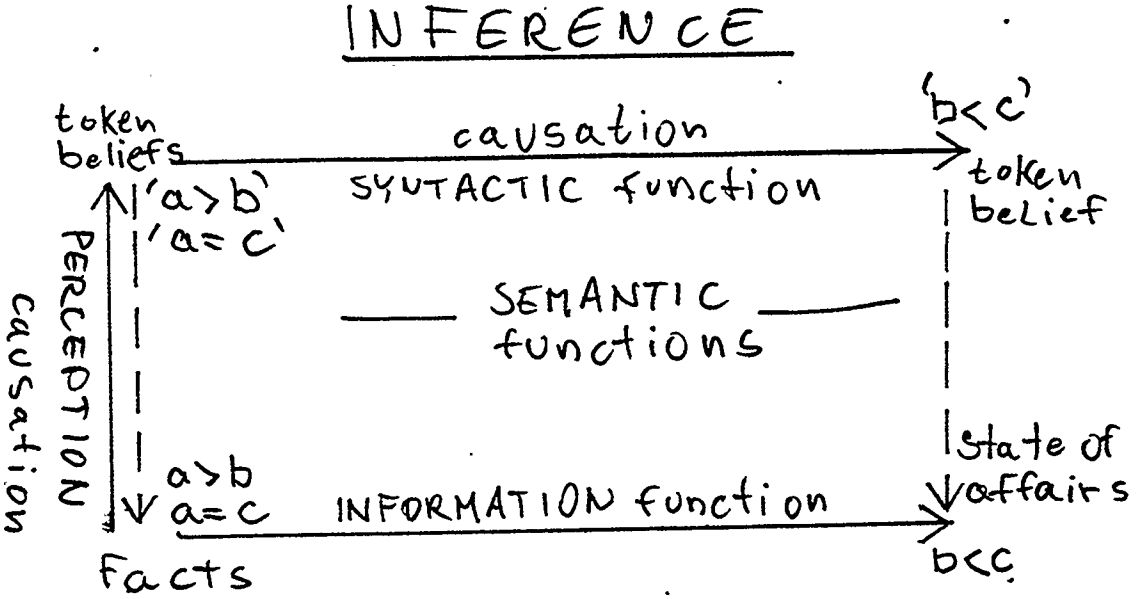


FIGURE 3.

These examples are of course valid and make their point clearly. What mystifies me a little is what extra information do they convey? i.e., does it not follow from the definition of computation outlined in the previous section, that it has to be actual as opposed to dispositional (i.e., in virtue of being causal processing of physical tokens)?

The rhetorical question is nonetheless posed once again.

"So when *is* a causal process a computation? I said in section 2 that it is when a semantic function makes the causally relevant properties of its stages syntactic. This implies that the semantics of computation is what generates its syntax. But it is not so obvious what generates its semantics: what makes a stage of a causal process actually embody information. So it is worth asking if I have got the interdependence of syntax and semantics the right way around. Perhaps information is embodied *because* it is processed, rather than *vice versa* But if so, there should be a criterion for a causal

process being syntactic that is independent of semantics. But is there?" (Mellor p. 55)

Once again, we already have the answer to the first question although it is here rephrased one more time (perhaps consistently with previous ones). The important question which is raised here is however quite clear and it is the central question of those computational theory of mind construals which - like Fodor's - take *symbol processing* to be an ineliminable part of the theory. To put it simply, the problem is that those symbols will have to mean something, whereas the symbols that the computer manipulates obviously do not mean anything to the computer itself. It seems on the other hand that if the mind does turn out to be a computational entity (and a symbol manipulating computational entity at that), those symbols that it manipulates will be meaningful to the mind. In other words, the question is (and the big task facing those who aim to construct an intelligent machine) whether the computer can be changed from a syntactic-machine into a semantic-machine, or - this is Mellor's point - can we even say that the computer is a syntactic machine, given that it is merely designed to manipulate the strings, not to interpret them?

The best known argument against computationalism constructed along these lines is probably the one due to the philosopher John Searle, popularly known as the "Chinese-room argument". It is has been rephrased and reproduced in countless volumes (e.g. Hofstadter and Dennett (1981), Searle (1992), Penrose (1989)). I choose to quote it from the transcript of a panel discussion that took place in 1984 at a conference on "Computer culture" between members of the "AI community" and other philosophers:

"Searle: The way I like to demonstrate the falsity of strong AI is to get you to imagine yourself instantiating a computer program for a certain kind of thought process. It's very important in these discussions to take the first-person point of view, to ask, What would it be like for me? Because that's what we know of being conscious and having thought processes. So imagine that there's a computer program for understanding Chinese, so that if you punch a question in Chinese into the computer, the computer can give out the right sort of answer. It has the right sort of database and the right kind of program so that it

can process questions in Chinese and give the right answers.

"Now imagine that you are the computer. You're locked in a room together with a whole lot of rule books for shuffling these Chinese symbols around. This will only work if you don't know Chinese. Like me; I don't know a word of Chinese. I don't know what any of these symbols mean. So there I am in the room shuffling these symbols around. The questions come in. I look up what I'm supposed to do when I get a squiggle squiggle sign and I go and match it with a squaggle squaggle sign. That is called a computational process over a purely formally specified element. These are what Simon and Newell call physical symbols; and now I am acting like a physical symbol system.

"Let's suppose these guys get good at writing the programs. I get good at shuffling the symbols. The questions come in, and I give out the right answers. One guy in responding to me said, > > Suppose one of the questions is 'Do you understand Chinese?' < < And I shuffle around - now I don't know what any of these symbols mean - and put out the symbol that says: > > You bet I understand Chinese. And how! What could be more obvious? Why do you keep asking me these dumb questions? < < What I want to say is that it's quite obvious, once you look at it from the first person point of view, that I don't understand a word of Chinese and I wouldn't learn Chinese from instantiating the Chinese understanding program.

"Why not? What is it that I have in English that the Computer doesn't have in Chinese? Notice that if I don't understand Chinese in that story, then neither does any other computer program understand Chinese, because the computer hasn't got anything that I haven't got in the story. What is it that I've got for English that the computer program doesn't have? Well I like commonsense answers. The difference is that in English, I know the meanings of the words, and in Chinese, I don't know the meanings of the words - all I've got is a set of formal symbols with a set of computational rules for manipulating the formal symbols." (In: Pagels (1984) p. 146.)

I have a number of things to say about the Chinese-room which are all relevant to Mellor's idea but I have to draw on arguments proposed by other authors quite extensively. I shall begin by a fact that may seem surprising, namely that the leading

theoretician of CTM, Jerry Fodor, can said to be in agreement with Searle along the main line of the argument.

4. Searle and Fodor

Searle argues that no computational theory will ever be able to solve the problem of the "intentionality" or the "essential aboutness" of conscious thought, the possession of which is usually acknowledged to be a basic fact about the mind. This, as it turns out, is not an argument against what is usually called the "computational theory of mind", that is Fodor's language of thought theory (LOT). The language of thought theory is constructed utilizing a kind of reasoning which I will have more to say about in chapter two (i.e. considerations of functionalism and physicalism). Although it is a theory of the nature of our thought processes, it *emphatically* does not offer any hints at what the mechanics of intentionality could be. In fact Fodor repeatedly stresses in his writings that (according to him) there is absolutely no strategy currently on the market that is a serious candidate for yielding an explanation of how intentionality comes about in the first place. A few quotes should suffice in this context:

"There are two, quite different applications of the 'computer metaphor' in cognitive theory: two quite different ways of understanding what the computer metaphor is. One is the idea of Turing reducibility of intelligent processes; the other (and in my view, far more important) is the idea of mental processes as formal operations on symbols. The doctrine of these essays [I am quoting from a volume of Fodor's papers. P.H.] is precisely that the objects of propositional attitudes are symbols (specifically, mental representations) and that this fact accounts for their intensionality and semanticity." (Fodor (1982) p. 24.)

Hence the "language of thought" hypothesis is a theory that builds upon evidence for, and asserts the symbolic nature of our thought processes, where the symbols which are actually manipulated (computationally) already have their interpretations. It is precisely

the interpretation part which LOT is not trying to explain and this is what Fodor means by the Turing-reducibility of intelligent processes. In a different work Fodor makes his views on the latter quite clear:

"We have, to put it bluntly, no computational formalisms that show us how to do this [i.e. Turing-reduce the intelligent processes. P.H.], and we have no idea how such formalisms might be developed. [...] If someone - a Dreyfus, for example - were to ask us why we should even suppose that the digital computer is a plausible mechanism for the simulation of global cognitive processes, the answering silence would be deafening" (Fodor (1983) p. 129.)

Of course I have already indicated what an answer to the question could sound like, and in the next chapter I shall examine a suggestion for providing a foundation for cognitive science along those lines. In fact I am rather puzzled why Fodor never mentions the connection between physicalism and algorithms (and I am also puzzled by the tone of finality), but then again, the emphasis is on the positive side of the theory[7].

There are two other major points I wish to make about the Chinese-room. One concerns the Turing-test and the conceptual possibility of the whole thought experiment, the other is a point about intentionality which supplements the above in some ways. I shall make the second point first.

5. Dretske and information

One could always argue that the trouble with the Chinese room scenario is that the room is a black-box in the sense that the mechanisms for generating the symbols (i.e.,

---

[7] Fodor also argues elsewhere (e.g. "Tom Swift and his Procedural Grandmother" in Fodor (1982)) against various other scenarios that are proposed to bridge the gap between computation and intentionality. In the paper I just mentioned it is argued for instance that compiler languages cannot substitute for the language of thought.

the characters of the Chinese alphabet) are not a *real part* of what is going on inside the box. Now, with an information processing system such as a computer this is not necessarily the case. There are many ways in which a computer system could interact with the environment and could form representations that are dependent on environmental factors. It could be claimed that *real* representations, such as are inside our head, derive their meaning precisely by their origin (say that our images are results of perceptual processes) and the fact that they are "used" in accordance with their derived properties accounts for their "intentionality". The suggestion is clear, even if the precise form in which the explanation is supposed to come from it, is vague. The most (at least to me) well known example for an AI experiment which provides weight for this line of argument is the robot named Shakey developed at Stanford in the sixties. Shakey could recognize objects (simple ones such as cubes and spheres) and move them around on command[8]. Although Dennett uses the example of Shakey to argue a different point it seems obvious to me that "he" (i.e., the robot) is in a situation where the symbols (i.e, the symbols that are on the communication screen) could be said to "mean" something if only to a degree. They're not merely acquired by fiat and transformed according to set rules as in the Chinese room, but they are in fact procedurally connected to the objects to which they refer[9].

---

[8] The whole experiment is described in Dennett (1991) p. 85-95. Also in Hofstadter (1979). Dennett points out precisely that what goes on in the "thoughts" of Shakey is not a recognizable representational process.

[9] This (and the discussion so far) also raises the question whether it would make sense to establish a distinction between representational-syntax of physical tokens and causal-syntax. What I mean by this is the following. In the 'ball-machine' example for instance, I would like to say that the balls that "represent" the propositions do not have representational syntactic properties merely causal ones. The physical properties of the tokens (the size of the ball) are such that upon feeding it to the machine it caused the right sort of output and the interpretation of the output was in fact dependent via an information function on the interpretation of the input. In this sense a physical property of the ball does represent a certain proposition and thus it had *causal* syntax. It is also true however that no physical properties of the balls are such that relations between them reflect any sort of relation between *semantic* properties of the propositions of which the balls are tokens. Even if it <u>is</u> a mystery how "...anything could be about anything else.."

Fred Dretske has argued in a paper entitled "Machines and the Mental" (Dretske (1985)) that no such approach will in fact suffice for providing an explanation of meaning. It will be useful to look at some of his arguments since he is directing his entire attention to our problem.

The goal of the argument is to justify an intuition - one which he shares with Mellor:

"I happen to be one of those philosophers who, though happy to admit that minds compute, and in this sense *are* computers, have great difficulty seeing how computers could be minded. [...] For machines, even the best of them, don't have an IQ. They don't do what we do at least none of the things that, when we do them, exhibit intelligence. And it's not just that they don't do them the way we do them or as well as we do them. They don't do them at all. They don't solve problems, play games, prove theorems, recognize patterns, let alone think, see and remember. They don't even add and subtract." (Dretske (1985) p 23. p 24.)

It is perhaps interesting that Dretske approaches the problem from the opposite side as Mellor. Whereas the latter claims that minds do other things besides computing, Dretske

---

(Fodor) that is, how symbols derive their meaning from the system that creates them, it is certainly true that some representational systems (and in fact Mellor above makes this point) are richer than my ball example in the sense that relations between intrinsic properties do *reflect* semantic relations between what the tokens represent even if they do not intrinsically refer. Such is the case of course with natural languages. The point is probably that there is no *necessary* relationship between the two kinds of syntaxes. The properties in virtue of which the tokens cause the right sort of things might be totally independent from the properties in virtue of which the tokens represent. (I take it that theories such as transformational grammar question this assumption. I gather that it is mostly assumed that tokens of the language of internal representation are causal by virtue of the same properties by which they represent and this is one of the reasons why it is possible to infer to the structural grammar (or deep syntax) from the natural language level.) This whole remark is strictly intended as a footnote but I feel that the idea of the distinction is sufficiently interesting to deserve a mention.

shall argue that machines never even begin to do anything minds do on a daily basis[10]. The argument is quite compact:

"The following is an attempt to show that whatever it is that computers are doing when we use them to answer our arithmetical questions, it isn't addition. Addition is an operation on numbers. We add 7 and 5 to get 12, and 7, 5 and 12 are numbers. The operations computers perform, however, are not operations on numbers. At best, they are operations on certain physical tokens that stand for, or are interpreted as standing for, the numbers. Therefore, computers don't add." (Dretske (1985) p. 25.)

Somewhat to my satisfaction Dretske adds : "In thinking about this argument (longer than I care to admit) I decided that there was something right about it. And something wrong." My satisfaction derives from the fact that I have also thought about the problem longer than I care to admit. The result of the process is the last chapter of this essay and it amounts to no more than a suggestion.

What is right about the argument, is - according to Dretske - that it points out what Searle and Fodor have pointed out, namely that pushing around representations of things is not the same as understanding them. What is wrong with it is something that I think can be fairly advanced against both Searle and Fodor, namely that if we do succeed eventually with eliminating homuncularities from the explanation of thought-processes then we will inevitably end up with something like Leibniz's mill with "...parts pushing

---

[10] Dretske makes an important distinction between kinds of tools. He points out that there are two kinds of tools that we use. With one kind of tool such as keys, we use the keys to open doors, but we do not say that 'keys open doors' (i.e., on their own account). With a tool such as a vacuum-cleaner it is not quite so obvious whether we or the machine "picks up the dust". The point in this context is that if computers are like vacuum cleaners then the CTM thesis might have a chance (i.e., if chess-programs actually *play* chess). The claim is that they don't.

against each other"[11]. And in fact there is a sense in which all the arguments against computationalism take us back to Leibniz's mill. Here is what Dretske says:

"This argument, as I am sure everyone is aware, shows too much. It shows that we don't add either. For whatever operations may be performed in or by our central nervous system when we add two numbers, it quite clearly isn't an operation on the numbers themselves. Brains have their own coding systems, their own way of representing the objects (including the numbers) about which its (or our) thoughts and calculations are directed. In this respect a person is no different then a computer. Biological systems may have different ways of representing the objects of thought, but they, like the computer are necessarily limited to manipulating these representations. This is merely to acknowledge the nature of thought itself. It is a vicarious business, a symbolic activity. Adding two numbers is a way of thinking about two numbers, and thinking about X and Y is not a way of pushing X and Y around. It is a way of pushing around their symbolic representatives.

"What is wrong with the argument, then, is the assumption that in order to add two numbers, a system must literally perform some operation on the numbers themselves. What the argument shows, if it shows anything, is that in order to carry out arithmetical operations, a system must have a way of representing the numbers and the capacity for manipulating these representations in accordance with arithmetic principles. But isn't this precisely what computers have?" (Dretske (1985) p 27.)

The answer to the latter question is, of course: no, and I will get to that in a moment.

---

[11] One of the differences between Searle and Fodor on this point is that Searle actually has a theory about how mental phenomena are biological phenomena just like digestion for instance, and in this sense we cannot expect an explanation which could be functional. Digestion is just the kind of thing that does not take place outside the organic world. (See Searle (1992) and in the discussion Pagels (1984)) Fodor of course flatly rejects this, because it contradicts a basic assumption of functionalism, namely that folk psychology describes information systems other than implemented in biological material. (On this, see chapter two of this essay.)

I have quoted Dretske at some length for I hope that what he says reinforces the intuition which I would like to have emerged from the discussion namely that if we are going to accept the modern version of Leibniz's argument in the first quote, then we will have to find an argument which we can substitute for what he says in the second quote, namely that the difference between artificial and natural machines is, that the latter are machines even in their smallest parts. For otherwise it does seem that the argument shows too much. It looks as if most arguments against computationalism (at least the kinds that I am aware of)[12] accept some sort of representational theory of mind, and once this is accepted, exactly the same kind of problems begin to emerge that Leibniz pointed out. Sometimes (as in Mellor or Searle or Penrose) the representational theory of mind is taken to be a theory about how computers *think* and Leibniz's argument is used to point out how in fact they do not. In other cases such as Dretske or Fodor, RTM is taken to be a theory about the mind and Leibniz's mill is used to point out how in fact there is no computational explanation for intentionality. In this essay I will look at only one alternative to this approach which will suggest a possible replacement for Leibniz's second remark. I have already mentioned connectionism (which attempts to make do without representations) and there are all sorts of neurobiological and physical theories which attempt to get around the problem but in general these are not well established and ususally contain one or more intuitions which are far from being commonly accepted. I think in fact that the attempt to be examined in the next chapter is the only commonsense alternative to RTM for vindicating at least the possibility of computationalism for the explanation of intentionality.

In the remainder of his paper Dretske offers an information theoretic approach (presented in detailed form in Dretske (1981)) for the partial explanation of intentionality. More precisely he separates the two problems. The arguments advanced are quite

---

[12] It could be an interesting project to figure out whether there is a principled way of categorizing arguments about CTM. I am thinking of something like Kant's categorization of the possible proofs for the existence of God. As arguments about CTM appear to be slightly antinomical in nature this is perhaps not so far fetched as it sounds.

straightforward. They underline the difference between the information content of symbols (which are readily explainable by an organism's or robot's interfacing with reality in the appropriate ways) and the meaning of symbols. The question is: how could one build a system to which the symbols that it manipulates mean something. Dretske's answer is that the above is how we start, but then we will never get as far as we would like. On the one hand this reinforces the AI thesis, thus it follows from the arguments that it is not a priori impossible to build a machine which will satisfy the Turing-test, on the other hand, however, mere information processing will never suffice for explaining mentation. The reason for this is, that there is no way around the fact, that in the final analysis it will be *meanings* which will have to figure in our explanations of truly intelligent mental processes as *causes*. Dretske gives an example of a marijuana detecting dog to pump the intuition that although the dog reacts in "appropriately behavioural" fashion it still doesn't have anything resembling beliefs about the marijuana. (It does follow more or less from Dretske's arguments that if the dog had beliefs it would have representations, but this is a dogma in the representational theory of mind anyway).

"The difference between machines (or dogs) and the agents who use them is that although machines (and dogs) can pick up, process and transmit the information we need in our investigative efforts (this is what makes them useful tools), although they can respond (either by training or by programming) to meaningful signs, it isn't the meaning of the signs that figures in the explanation of why they do what they do. Some internal sign of marijuana, some neurological condition that, in this sense means that marijuana is present, can cause the dog's tail to move, but it isn't the fact that it means this that explains the tail's movement. This, I submit is the difference between the dog and its master, between the machine and its users, between the robot and the people that it replaces." (Dretske (1985) p 31.)

One feels a slight tone of hesitation in the text when Dretske draws the final conclusions, which is not surprising for he would certainly like to avoid both "mind-stuff" theory and letting the audience leave empty handed. The extra twist to

information theory is that we should also postulate "feed-back" mechanisms which play a role in the determination of the meanings of symbols.

"For such systems [i.e. those possessing feedback mechanisms. P.H.] the internal signs not only have meaning, this meaning affects the way the system manages these signs; and it is in this sense that the signs mean something *to* the system in which they occur." (Dretske p. 32.)

Dretske of course realizes that this is still insufficient to account for intentionality but I feel he is very reluctant to let go of the possibility of some functional argumentation. In the closing paragraph however, he moves very close to Searle's final position (a version of biologism).

"...we, but not the machines and a variety of simple organisms are genuine thinkers of thoughts. What gives us the capacities underlying the difference is a long and complicated story. It involves, I think, issues in learning theory, our multiple sensory access to the things we require to satisfy our needs, and the kind of feedback mechanisms we possess that allow us to modify how we manipulate internal signs by the kind of results our previus manipulations have produced. But this, clearly is a story that we expect to hear from neurobiologists, not from philosophers. All I have been trying to tell is a simpler story, a story about the entrance requirements to the club." (Dretske (1985) p. 32.)

In the next chapter we will in fact hear a story from a philosopher, not a necessarily correct one, I submit, but a story nonetheless. And it is important in the sense I have already mentioned - it will allow us to go beyond the Leibnizian paradigm of argumentation.

6. Passing the Turing-test

Now to move up a level in this nested loop, and back to the Chinese-room once more. The latter is certainly a very elegant argument, and this extended look at Fred Dretske's paper was intended to show that an attack from the information-theoretic position is not likely to succeed (although it is one of the most likely places to start from). In other words, it seems that even if we allow for some connection between the room and the external world it will not change the situation much. Then of course, another problem with the scenario is that as it is set up, with the English speaker locked up in the room shuffling the symbols according to rules, it is difficult to see how the thought experiment as a whole could be brought closer to life in a way so that Dretske's arguments could begin to seem relevant. And indeed this points to a general fault with the argument, which I happen to think is a major problem, namely that it is entirely unrealistic. Thought experiments should at least be nomologically possible, and the Chinese-room does not seem to be so.

The Chinese-room passes the Turing-test "big-time". In fact this is a basic assumption of the experiment. This does not seem to be the result of carelessness on Searle's part, remember, he says: "..Let's suppose these guys get good at writing the programs. I get good at shuffling the symbols." The problem with this is that obviously the symbol shuffling skills of a distinguished professor from California are not relevant to the workings of the Chinese-room, whereas getting good at writing the programs is very much so. It just seems that the size of this particular "intelligence loan" (a phrase of Dennett's) is just a stretch bigger than what we should be prepared to to grant. In fact, Fodor - toying with the idea of a general question-answering device - says the following in the context of discussing the vacuity of some functional explanations:

"Still worse: the appearance of functional explanation can be sustained even where we know - on independent grounds - that nothing could perform the function specified. Here, for example, is a theory of the (occasional) human capacity to provide true answers to questions posed: Inside everyone's head is a universal question-answering device. The class of such devices is functionally specified as follows. A universal question-answerer

is a function from questions unto their answers; given a question as input, it provides the answer as output. [...] Now this story doesn't even manage to be true-but-question-begging. It has got to be false, since there couldn't be anything with the input- output characteristics that the universal question-answerer is alleged to have. >>Is a device which takes each question onto its answer<< doesn't, as it were, succeed in being a functional definition. [...] Something appears to have gone wrong." (Fodor (1982) p. 12.)

It seems to me that this is an excellent description of the Chinese-room experiment and what appears to have gone wrong is precisely as Fodor says, that there is no provision made for an actual mechanism that could do what the universal question-answerer or the Chinese-room does. Fodor in fact proceeds to introduce Turing machines in this context as devices which provide for mechanisms. I hope that by now it should be apparent that there are problems with this and in this sense Searle provides for a mechanism as well: the written program i.e., the instructions that the shuffler follows[13]. If such a program could be written, it would pass the Turing-test and at the same time the distinction between CTM and AI would have to fall. Of course, as the distinction now goes, CTM claims that there are some things (i.e. kinds of mental phenomena) whose "actual" presence is not necessary for the solution of an arbitrary task involving intelligence. In other words the theory of mind that is presupposed for CTM rejects behaviourism (see Chapter 1.). What I'm trying to point out is that in arguing for the truth of the AI thesis, arguments quite often substitute superficial behavioural criteria for mental events where the principles of simulation are quite transparent. (A computer with a greeting prompt.) The problems with behaviourism are numerous (See Johnson-Laird (1988) for a short but incisive introduction in the context of CTM), among them is the phenomenon that there is generally an infinite disjunction of behavioural responses paired to what seems to be

---

[13] As far as empirical evidence goes it is almost certain of course that such a program cannot be written (failed attempts at creating a natural language translator are attempts that failed at an easier task, and the moral gleaned is worse than disheartening). See Hagueland (1985) on Bar-Hillel).

a single mental state (such as a desire). (This is in fact something to be said in support of long term relationships.) The Chinese-room however appears to be able to simulate *all* the behavioural manifestations of a being endowed with mentality (verbally at least). And although for reasons just pointed out it is not easy to argue positively *for* behaviourism it seems just as difficult to argue negatively against it. It seems, in other words, that if a program such as the one Searle installed in the Chinese room is nomologically possible then the least we can say is, that intelligence which is perhaps differently constituted than human intelligence (but the same in all other respects) is possible as well. And even more importantly, in the final analysis we will have to concede that the Chinese-room *taken as a whole* does in fact understand Chinese. To insist that it doesn't understand Chinese but *no one will ever be able to get it to show this lack of understanding* would be a truly odd position to take[14].

7. Mellor's "own"

I hope that this extensive digression into the kinds of arguments and projects that are parallel to Mellor's will serve to place what he has to say into perspective. The moral so far is that there are no *significantly new* suggestions in his approach to the problem of computationality and in fact I don't think that there are any to be found in the rest of the paper. I shall take a look at parts of it anyhow because they touch on some important issues.

We left off at the problem which Mellor stated as the primacy question between syntax and semantics. Compared with the other arguments (Dretske, Fodor and Searle) he is

---

[14] What I mentioned in the first chapter about the difference between CTM and AI imply, that the latter is directed at eliminating arguments involving partiality, which is quite important. Searle, in fact, sees the point, that if the Chinese-room would only be good at particular tasks, then the argument shows nothing apart from what people working in AI already know, namely, that the best programs anybody can even conceive of as we speak may be able to solve difficult problems, but they won't be able to *actually think*. Searle has to posit that the room thinks in order to demonstrate that it doesn't. If what I said sounds like a 'common-sense' argument...well...

trying to offer something by way of a definition of "syntax" which would be independent of semantics. This is of course intended to be a speculation on how to transcend the symbol-shuffling predicament of the guy in the Chinese room, that is, it is an attempt at endowing the shuffler with understanding. We have seen that the definition is far from being clear. I have also mentioned that there is a problem about whether it is what I called the causal-syntax or the representational-syntax that Mellor is talking about. One could in fact conceive of the causal-syntax as being independent of semantics, although as I said before, if there is no necessary relation between the causal-syntax and the representational-syntax then whether a system has an independent causal-syntax would say nothing about its semantics.

The final answer to the question whether syntax could "precede" semantics is of course a resounding "no":

"The syntax of computers is after all a product of their semantics. Only when we know where the semantics come from will we know what it takes to be a computer, and how much of the mind may be one. [...] It is in fact obvious, once we think of it, where the semantics of computers come from: they come from us. The computers that have prompted and given sense to our question compute because we compute with them. Computers may, in the future, compute for themselves. But at present they compute for us, and represent what we use them to represent." (Mellor, p. 57.)

It all boils down to a discussion of the good old question about the origin of "holy" intentionality. The puzzle remains a puzzle.

Before laying down arms however, Mellor dismisses an interesting suggestion. There are two ways - he suggests - in which syntax could "come about" without semantics. One could attempt to say that a, a system is syntactic in virtue of its being a rule following as opposed to a natural-law-abiding system or b, "A process is syntactic when, like a Turing machine, it embodies an *algorithm* that can take input, and yield output, of unlimited complexity." As I said, both possibilities are dismissed, the second, perhaps not entirely justifiably.

The problem which - given the assumption of physicalism - will always remain is, to emphasize it just once more: Leibniz's problem. If it's not a machine then what is it? Leibniz's answer is the Monadology, and the weird doctrine about every part of a natural machine's being a machine. Our answer has to be: it is still "some sort of machine". What kind of machine? It seems, that if we hold on to the complex of assumptions that has been dubbed "the representational theory of mind" (or computing), then the answer will be: it is certainly not a computer. The point I am trying to make is that if one is truly committed to a non- computational explanation then "algorithmicity" will also have to be dismissed for reasons to be elaborated in the following chapters. It seems that according to physicalism there has to be something about the brain which is syntactic and which will in fact "precede" semantics if in no other, then in the evolutionary sense and that something will have to be looked for in a wide sense of algorithm (something like a discrete state-like organization). Anyway: the mark of being syntactic seems to be algorithmicity, especially if we tie "syntax" to tokens. More on all this soon.

The upshot of the notion of computation developed in the paper is thus that any mental process which cannot conceivably be taken to be the processing of propositions in the way schematically shown in Fig. 2, cannot conceivably be computational. There is an extensive and interesting discussion in the paper on how *we* use computers to make inferences and also how inference as we do it could be explained computationally. Whether it will in fact be explained computationally is something which the author takes to be an empirical question. (And there is evidence for it. See Mellor p. 61. and Fodor (1975) scattered.) In addition to inference, Mellor also concedes to the possibility of certain (high level) mechanisms of perception turning out to be computational. (sections 5 - 7) The important part comes however in the final section of the essay:

"How much computing there is in perception is therefore not for me to say: maybe a little, but maybe a lot. And inferring is computing by definition. But there is more to mental life than perceiving and inferring, and none of the rest of it, I shall now argue, is computation." (Mellor p. 64.)

The rest of the mind's activity turns out to consist of two kinds of things, processing sensations and processing other kinds of propositional attitudes than beliefs (desires, hopes, fears etc). The arguments against these being computational is not a very good one on the one hand, but it does tell us something about their nature. It is here that Mellor uses the definition of computing to its full extent, and he does it in a way that is a bit disappointing. When we began, it looked as if the paper was going to work with an unusually liberal conception and now it turns out that the only reason why neither sensations nor other propositional attitudes are computational is because they do not count as the "processing of information".

"Many mental processes are not computational because their stages are pains or other sensations which represent nothing. If for instance a loud aural sensation S gives me a headache H, that is a mental process. But only the tokens S and H are processed: *no information is*, because nothing is represented." (Mellor p.64.)

I take it that it is fairly obvious that this is a bad argument[15]. Even if we accept that "machines cannot feel pain" which is not a necessary truth precisely because of the fact that *we* do feel pain, one could still argue - as a good materialist - that pains do in fact represent something, i.e. they represent the physical processes which stimulate the nervous system in a particular way. That is: the sensation of pain represents the actual pain. This is certainly not an argument about the Leibnizian question, that is whether machines with minds are possible. All Mellor says is that feeling pain is not like inference. This seems to be the problem with the other propositional attitudes as well.

"The causal processing of propositions embodied in tokens of all attitudes must be syntactic. But why then call it "computation" only when the attitude is a belief? [...]

---

[15] Although the real issue here is the difference between functional and qualitative states of mind. The problem is a central one in the philosophy of mind. I shall come back to it in the next chapter.

What makes the processing of belief so special? [...] The short answer is that processing tokens of other attitudes is not processing *information*. Pieces of information are all propositions: but not all propositional attitudes embody information. Only beliefs do. Believing in P does embody the information that P: wanting, hoping for and fearing P do not. That is why inferring is processing information, i.e. computing, and the processing of hopes, desires and fears is not." (Mellor p. 65.)

The very simple way around this problem is what Fodor calls "being in the right computational relation" to a certain proposition. Mellor concedes that processing of other attitudes besides beliefs *could* be syntactic (which is obviously the case), only it isn't going to be the same kind of thing as simply processing a proposition, hence it won't be computation. The problem with this is (and this is what Fodor means) that it assumes that the only kind of computational relation a computer can have to information is storage in memory. Thus we could (metaphorically) say that a computer has a certain belief if it stores in memory a token of that belief. However, it seems to me that processing a token which is in the memory *in different ways* is precisely what could account for the "modality" of the token and counts as qualifying the relation of the machine to the token. This state of affairs has been described as putting the token in particular boxes labeled with names of the propositional attitudes. Thus if a proposition is in the "fear" box then the proposition will take part in the processing as a fear, if it is in the "hope" box, then it will participate as a hope. This seems to be a straightforward way of meeting the requirements of functionalism. And the big question is whether this is not exactly the way in which propositional attitudes are individuated as kinds. In other words, this is an empirical hypothesis about the nature of our psychology and is certainly not to be dismissed on a priori grounds. Mellor of course does not dismiss it and he is right that processing other propositional attitudes is different from processing beliefs and he might also be right in saying that according to his definition of computation, this will not be computation. The problem is obviously with the definition of computation.

There are two ways according to Mellor in which it would be possible to get around the problem. One is the decision-theoretic model where it is assumed that the agent

actually computes expected utilities for actions based on propositional attitudes.

"Like the computational mechanics of section 3 [See figure 3. P.H.], the idea makes perfectly good sense only: it's obviously false. I could act by using decision theory to work out how to act and then acting that way. But I don't; any more than a mass M uses Newton's laws to work out how to react to applied forces. [...] The objection to such computational theories is not that they are nonsense, or vacuous, but that they are false. No one thinks the truth of a computable theory of mechanics makes M a computer, because we know M can't compute. But we can: hence the tempting idea that what makes a computable theory of mind true is that we compute with it." (Mellor p. 67.)

There is no argument here besides an appeal to introspection and the final conclusion is rather disturbing. It turns out that the argument rests on a rather arbitrary differentiation between kinds of processing, calling one computation and the other a mystery.

The very last issue touched on in the paper concerns the possibility of reducing all propositional attitudes to beliefs and hence fulfilling the condition of computation imposed by Mellor. This is not a priori impossible considering the fact, that all propositional attitudes are *propositional*, i.e., are relationally dependent on a proposition which in turn can be construed simply as stored beliefs. The problem however is quite complex and it seems that one minimally has to assume desires in order to attempt cutting the number of attitudes by reduction. Whether desires and beliefs are sufficient is an interesting question. I feel they are not and it has been argued - in connection with emotions - that gaps remain in folk psychological explanations if one doesn't allow for a wide range of modalities, in particular emotional modalities. One of the reasons for this is that the structure of emotions seems to be complex in ways that preclude purely functional explanation. (See Gordon (1987) and Zajonc (1980) for a case against and Green (1992) for a case for parsimony in intentional explanations.)

In the next chapter I shall turn to a suggestion for a subsymbolic computational foundation of cognitive science.

CHAPTER III.

A "different" approach

1. The idea

The topic of this chapter is the idea that what is wrong with Leibniz's second thought is that it is superfluous. There *is* no essential difference between nature's machines and artificial ones and in fact the computer - the ultimate machine - is the right model for all natural processes. The ghost of this thought is one that haunts not only cognitive scientists of all persuasions[16], but also physicists and other natural scientists. Arguing this position conclusively is beyond the scope of this essay. However, I will discuss in detail a typical manifesto of the belief, a paper by David Chalmers[17] in which he proposes a computational foundation for the study of cognition. These ideas provide ample illustration for the point I am trying to make, and they also lay bare some of the methodological prejudices that are involved in other theories of the same sort.

One source of the problems which are peculiar to Chalmers' work is, perhaps, the very grandness of the scheme, and the corresponding generality of the ideas involved in it. Chalmers' concern in the paper is to give a simple foundation for cognitive science, which he takes to be equivalent to establishing the truth of two theses dubbed: (a) The thesis of computational sufficiency and (b) The thesis of computational explanation. There is a certain bluntness and immediateness characteristic of the tone of the whole

---

[16] E.g. see the discussion about 'biological machines' in Pagels (1984). Nobody mentions whether perhaps it is the notion of a machine which causes the havoc.

[17] David Chalmers: "A Computational Foundation for Cognition". The paper is available through the INTERNET by "ftp" from Washington University, Saint Louis. Unfortunately the nature of the manuscript prevents me from giving exact page-references. I shall try to indicate the location of the quotes by markers such as 'Chalmers then remarks...' or 'In an earlier section, where he said...'

essay which is apparent already in the formulation of the theses.

"First, underlying the possibility of artificial intelligence there is a thesis of *computational sufficiency*, stating that the right kind of computational structure suffices for the possession of a mind, and for the the possession of a wide variety of mental properties. Second, facilitating the progress of cognitive science more generally there is a thesis of *computational explanation* stating that computation provides a general framework for the explanation of cognitive processes and of behaviour." (Chalmers (1994))

A footnote serves to clarify the goal of the essay somewhat further. The "accounts of implementation" Chalmers refers to are theories about what it means for a physical system to be computational:

"It is surprising [...] how little space has been devoted to accounts of implementation in the literature in theoretical computer science, philosophy of psychology, and cognitive science, considering how central the notion of computation is to these fields. It is remarkable that there could be a controversy about what it takes for a physical system to implement a computation at this late date." (Chalmers Ibid.)

In Chalmers' opinion this is remarkable because settling the question once and for all is likewise, remarkably simple. I shall now say a few words about how Chalmers fits into the general framework of the first two chapters but I will not go into any details as yet.

The reader will remember that in the introduction to the essay, I had differentiated between two different kinds of intuitions underlying the relevance of computers to the AI thesis. The first had to do with the phenomenology of the computers - the kind of things they are capable of - and a deepened form of this first reason turned out to be basically the founding intuition of the kind of computational theories of mind (i.e. the "representational theories of mind") which were made the subject of scrutiny in Chapter II. (e.g., Turing-machines providing causal mechanism for the realization of

transformations in representational systems depending on representational syntax). The other reason had to do with metaphysics, namely the general assumption that the explanatory physics of the mind (especially as far as explanation of intentionality goes) has to make do without homuncularities and what has to replace homoncularities are precisely algorithms[18]. How this is supposed to come about - according to quotes we had from Mellor, Fodor, Searle and Dretske - is not for the philosopher to answer and has nothing to do with what is usually assumed under the label of computational theory of mind. This is the assumption Chalmers is questioning, and the reason is what I referred to in saying that the computer is in some sense the ultimate machine. Fodor and others take this to imply that the computer is the right model for "syntactic" processes. Chalmers goes a bit further and claims that the computer is in some sense the right model for all natural processes which realize some sort of "function" or have some meaningful "stability" to them. To put it differently, the usual computational theory of mind is an attempt to answer Leibniz's first objection but (as chapter 2 demonstrated) its success is controversial[19]. This theory also has nothing to offer in the way of replacing Leibniz's second argument (i.e to say what it is about a natural machine that allows for mind, which no artificial machine could duplicate); it merely asserts that it has to be wrong (i.e., in the sense that some mental phenomena are mysterious because they are unexplained, not because they're unexplainable.) Chalmers claims that Leibniz's second argument is unjustified, for computation does offer an explanatory framework for intentionality.

---

[18] The metaphysical assumption is that any "meaningful" non-homuncular (& non-teleological) material process is necessarily an algorithmic one. That this is a basic assumption in the kind of theories of mind I am dealing with here (i.e., non- dualistic and non-mystical methodological ones) is something that I am arguing for, not one that is explicitly assumed by anyone.

[19] I am of course not criticising "computational psychology" in any detail. Thus much of Fodor (1975) for instance is devoted to speculations and arguments about what kind of evidence there is for the symbolic nature of particular mental processes. In this essay all I am interested in is the nature of the assumption labeled by the adjective "computational".

Chalmers' idea, in a nutshell, is that "functional" properties of physical systems depend only on the "causal organization" of the physical system and that "causal organization" is nothing but a realization of a certain computation. As mental properties are functional properties of a physical system, it follows as the night the day that they are also computational properties.

Perhaps the biggest problem with Chalmers' presentation is that we never get an account of how exactly psychological properties are supposed to be functional properties. Here's what Chalmers tells us about this:

"Psychological properties, as has been argued by Armstrong (1968) and Lewis (1972) among others, are effectively defined by their role within an overall causal system: it is the pattern of interaction between different states that is definitive of a system's psychological properties. Systems with the same causal topology[20] will share these patterns of causal interactions among states, and therefore, by the analysis of Lewis (1972), will share their psychological properties (as long as their relation to the environment is appropriate)." (Chalmers Ibid.)

Essentially the whole argument is contained in this quote. What makes it difficult to argue with Chalmers' position is, as I pointed out above, that he never actually tells us what is meant by stating that "...psychological properties are effectively defined by their role within an overall causal system." The way in which this is put in fact, could hardly be more annoying. Notice that Chalmers says that Lewis and Armstrong *among others* have argued for this. Now, even if they had argued "this", they could have been wrong, but as a matter of fact, they argued for no such thing (not, at least, for the second part of the sentence). It is a feature of the way Chalmers places this quote that if someone wants to disagree with him, he is committed to an extensive discussion of the literature. For Chalmers does not even state his position, what we get in this quote is all he will

---

[20] For the notions of "psychological property" and "causal topology" see sections 3. and 4. For Chalmers' references, see my bibliography.

say, namely "psychological properties are defined by their role within an overall causal system". The conclusion is supposed to follow from this, and this alone.

Luckily, however, what Chalmers is referring to is a view of 'mental properties' which is commonly summed up under the title of "functionalism". Whoever is familiar with the literature in the philosophy of mind will readily agree that it is very unlikely that there is anything behind this particular thought of Chalmers which could possibly amount to anything deeper than what basically counts as folklore among philosophers versed in the philosophy of mind literature. Given this assumption I will proceed by relying upon Fodor's description of functionalism (to be found in Fodor (1982)) in discussing Chalmers' claims. I feel justified in this because, if anybody, Fodor should certainly be included in the phrase 'among others', and providing conclusive evidence within the limits of this work that the references Chalmers gives contain the same kind of things as Fodor's writings, seems impossible. David Lewis' paper is quite technical and Armstrong's work is a long book. The reader will have to take my word for it that I have checked, although I think it will become apparent from the arguments that nothing crucial hinges around textual criticisms. My statement above, that Chalmers' assumption about them is incorrect will be supported by my showing that "functionalism" as described by Fodor does not assume what Chalmers' takes it to assume, and they in turn are functionalist. As I said, this should suffice for anyone who "knows what's going on". I take it that my argument is still more honest than Chalmers'. All he says - to repeat it once more - is that psychological properties are defined by their role in a "causal system". I will assume that this actually means something and I shall also attempt saying what it means.

In what follows, I shall first present and explain Chalmers' theory of implementation. Throughout this section I will assume that a functional property of a physical system is to be understood "roughly" as a property which can be defined by reference to the "causal role" it plays in the physical system's interaction with the world. An example of a functional property I will work with is the property of being a household machine such

as a coffee-maker. This seems to be a functional property. One can define coffee-maker as a physical system which causes water and ground coffee-beans to turn into liquid coffee. Hence, the machine is defined by the kind of causal role it plays. One of the key question will be of course, whether a 'psychological property' is the same sort of thing or not. For the moment, let's assume that we know what a functional property is, and that a coffee-maker is an example of it.

2. Another definition of "being computational"

Here's the basic version of the definition:

"A physical system implements a given computation when the causal structure of the physical system mirrors the formal structure of the computation." (Chalmers Ibid.)

In a little more detail this comes to:

"A physical system implements a given computation when there exists a grouping of physical states of the system into state types and a one-to-one mapping from formal states of the computation to physical state-types, such that formal states related by an abstract state-transition relation are mapped onto physical state-types related by a corresponding causal state-transition relation." (Chalmers Ibid.)

To understand this definition one has to have a basic knowledge of the theory of automata. As Chalmers elaborates his definition in terms of Finite State Automata, I will say a few words about what those are.

A Finite State Automaton (FSA) is a simple mathematical model for a certain class of computations (in fact the model itself defines a certain class of computations). It is simplest to picture an FSA as being a 'mindless robot' which can be in any one of a finite number of states and can take a finite number of inputs (react to a finite number

of kinds of stimuli) and can give a finite number of outputs. What an FSA "does" is accept input and give outputs and change states. Given that it is in a certain state, upon receiving an input it changes its state (could in fact "change" to the same state i.e., do nothing) and gives some output. The definition covers FSAs without outputs, namely the FSA which has nothing as output for any input. The adjective: 'mindless' is appropriate in this case, because the essential difference between an FSA and more sophisticated models of computation is that the FSA does not have memory, i.e., its behaviour is not dependent on the history of its past behaviour. It is not entirely correct to say this of course. "Memory-like" behaviour could be hard wired, but still, an FSA has none of the storage capacities which a Turing machine does[21].

FSAs are sort of boring of course but so are most "finite systems" which they describe, such as various household appliances. Simple machines which can be thought of as FSAs include anything from a coffee-maker to a television set. Given the right kind of input (i.e., filling it up with coffee and water and plugging it in) the coffee-maker will proceed to make coffee[22]. The beginning state is: no coffee, the endstate: lots'a coffee. The television-set can be either off or on (input: push the button) what is more it can even be set to different channels (push a different button). As a preliminary intuition, I expect the reader to feel an intuitive difference between the Drip'o Matic and the TV-set. Saying that the latter is in a particular state sounds somehow "more appropriate" to my ears then in the case of the former. What is more, I also feel that it is still not the right

---

[21] Figuring out simple limitations and capacities of different models of automata like FSAs and Turing-machines is an amusing pastime. It is not at all a trivial matter to see what and how these machines can do or how they differ from each other. Informal introduction to automata can be found in textbooks on cognitive science such as Johnson-Laird (1988) or in popular texts such as Penrose (1989) or Hofstadter (1979).

[22] Whoever has read the previous chapters and is now doing me the honour of browsing through this one could get hung up on this. After what I've said about the importance of Dretske's tools, this is just obvious loose-talk. In this case however it is not really important whether it is the machine making the coffe or not. As I said what is important about Chalmers is his proposed answer to the second challenge of Leibniz.

way to describe the TV-set. The closest model I can think of right now, which could count as a real life approximation to an FSA would be an elevator. Depending on the input to its console panel it gives as output successive stops at the sequence of floors that are punched in; and which floor it is going to visit next does not depend on which floors it has visited in the past. Much of the discussion about what is suggested in Chalmers' paper is centered around the question whether all these "thought-games" are illuminating in any serious sense or just aids for understanding - Hegelian ladders one throws away having planted a foot firmly on the ledge.

Here's an example of how Finite State Automata are usually depicted:

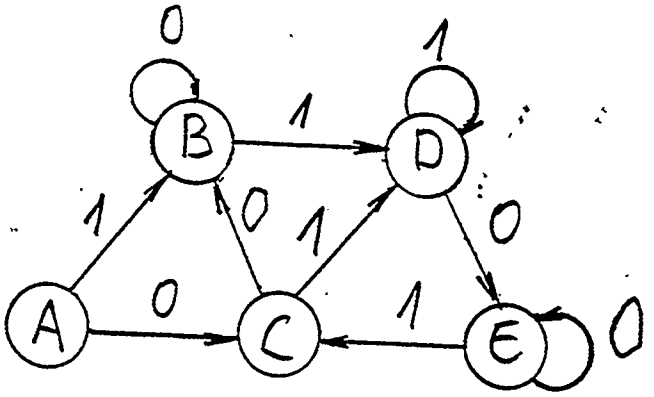


FIGURE 4.
An example of a Finite-State Automaton

The little blobs are symbols for states (the letters are their names or if you like the names of the outputs the machine gives upon arriving at the state) and the numbers on the arrows stand for inputs to particular states. The presence of a circular arrow indicates that for that particular input, the automaton just stays put, i.e., does not change states.

There is usually a beginning state and an endstate because there is usually a beginning or an end or both to everything (jokes aside, endstates are necessary if the machine is to

be used for recognizing strings of symbols for instance, i.e., it will have to indicate that it had finished computing). In the case of a machine like an elevator, this is superfluous: a trip could start from any floor.

The most important thing to notice is that the Finite State Automaton is a deterministic machine. Given a state and an input it has no choice any longer about what to do, the successor state is determined. This fact is what makes an attempt at defining implementation - such as Chalmers' - possible[23].

I would like to avoid getting bogged down with technicalities, although I feel it necessary to indicate that there *are* important technicalities in abundance, and this treatment does not even begin to sratch the surface of what could be said about automata.

Given the definition of an FSA the definition of implementation is now straightforward according to Chalmers:

"A physical system **P** implements an FSA **M** if there is a mapping f that maps internal states of **P** to internal states of **M**, inputs to **P** to input states of **M**, and outputs of **P** to output states of **M**, such that : for every state-transition relation $(S,I)\rightarrow(S',O')$ of **M**, the following conditional holds: if **P** is internal state s and receiving input i where $f(s) = S$ and $f(i) = I$, this reliably causes it to enter internal state s' and produce output o' such that $f(s') = S'$ and $f(o') = O'$." (Chalmers Ibid.)

The definition is basically a simple way of formalizing the intuition about household machines. In the case of the coffe-maker the beginning state of the physical system is the coffee-maker filled with water and coffee and an empty pot. Plug it in and the pot gets filled. We have thus defined an isomorphism between the causal state-transition structure of the physical system and a two state finite automaton (see Figure 5.a). The physical system turns out to be implementing a certain computation. Now, of course, the fact that

---

[23] There are models for computation such as non- deterministic automata (and probabilistic automata) where this requirement is relaxed. These are important in their own right, and there are philosophically relevant things to say about them, but not in this context.

the physical system implements this computation is *insufficient* for it to be a coffee maker. To give a literary example, Eeyore's birthday present in A.A.Milne's Winnie-the-Pooh implements precisely the same two state finite automaton. The balloon-shreds can be either in the empty honey-pot or outside it and this possibility of changing the state of the physical system is what causes Eeyore's delight. (See Figure 5.b) On the other hand neither the coffee-maker, nor Eeyore's present could be what they are *without* implementing this particular finite automaton. In other words implementing this particular finite automaton is a necessary but not a sufficient condition for a physical system for possessing a certain functional property.

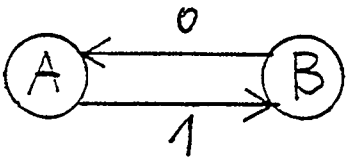


FIGURE 5.a
A two-state finite automaton

FIGURE 5.b
Eeyore's delight

So it seems to be the case for any system described in terms of implemented FSAs. The reason for this - as Chalmers remarks - is that the states referred to in the definition are "monadic"; they could be states under some very high level of description (such as the coffee-maker). Chalmers plans to remedy this defect of the theory by introducing so called Combinatorial State Automata (CSA), a generalization of FSAs and a formalization sufficiently powerful to describe Turing machines and other systems which are powerful enough to compute the effectively computable functions.

Chalmers argues that whereas implementing an FSA is merely a necessary condition for possessing functional properties, implementing a particular CSA is a necessary and sufficient one. Here's what CSAs do in Chalmers' words:

"The condition under which a physical system implements a CSA are analogous to

those for an FSA. The main difference is that internal states of the system need to be specified as vectors, where each element of the vector corresponds to an independent element of the physical system. A natural requirement for such a 'vectorization' is that each element correspond to a distinct physical region within the system, although there may be other alternatives. The same goes for the complex structure of inputs and outputs. The system implements a given CSA if there exists such a vectorization of states of the system, and a mapping from elements of those vectors onto corresponding elements of the vectors of the CSA, such that the state-transition relations are isomorphic in the obvious way. The details can be filled in straightforwardly as follows.

A physical system **P** implements a CSA **M** if there is a vectorization of internal states of **P** into components $\{...s_i...\}$ and a mapping *f* from the substates s_j into corresponding substates S_j of **M**, along with similar vectorizations and mappings for inputs and outputs, such that for every state-transition rule $([I_1,...,I_k],[S_1,S_2,...]) \rightarrow ([S'_1,S'_2,...],[O_1,...O_l])$ of **M**: if **P** is in internal state $[s_1,s_2,...]$ and receiving input $[i_1,...,i_n]$ which map to formal state and input $[S_1,S_2,...]$ and $[I_1,...,I_k]$ respectively, this reliably causes it to enter an internal state and produce an output that map to $[S'_1,S'_2,...]$ and $[O_1,..,O_l]$ respectively" (Chalmers Ibid.)[24]

What this framework suggests is that a computational specification of a physical system could be sufficiently detailed as to specify functional properties. The intuitive reason for this is that if a computation that a particular physical system implements is sufficiently complex, then chances are, that if another physical system implements the same computation, it will probably have the same functional properties. Thus, we should be

---

[24] I should remark that I am not even sure whether what Chalmers wants is *technically* possible. Thus it seems to me that the only way in which one can turn a Turing-machine computation into a CSA is by coding the input into the computation. In other words, there is probably no such thing as a CSA description of a Turing-machine taking inputs and giving outputs. This might not be an important point though; and the other points are worth making even if it is.

able to go down "deep enough" in the physical system, so that the implemented computation becomes detailed enough to individuate the system.

Now one of my problems with this is that I have no idea how one could slice-up the complex state of being a coffee-maker (or Eeyore's "balloon & pot") into "parts", the state-interaction between which will be enough to guarantee that no other physical system implementing the same interaction will be able to avoid possessing the same functional property. The intuitive reason behind this is, that after "slicing" to some depth we will encounter parts whose states are no longer functional (such as the state of being a sieve or a pipe for instance). The following short thought experiment is intended to be a "proof" that what Chalmers suggests is impossible.

It is a fact that a common personal computer, such as the cloned IBM PC sitting on my desk, can be programmed to implement any given CSA. That's what it is designed for, as a matter of fact. Now, suppose that there exists a CSA, the implementation of which makes a coffee-maker a coffee-maker. It is possible to write a program which will make my desktop computer implement the same computation. However I still won't be able to make coffee with my computer. Hence Chalmers' claim is false. Q.E.D.

It seems to me that this proof is completely sound, but I can imagine some objections which could be raised against it. Two of the objections in particular seem important.

Someone could object that writing a program merely specifies an "abstract" implementation of the CSA in question, whereas the definition requires a physical system with parts whose physical states implement the same computation.

The objection is not valid however, for the computer <u>is</u> a physical system and there is nothing to prevent someone from identifying the physical processes that underlie running the program. It is no good insisting that the computer is still a different physical system from the coffee-maker - hence, obviously implementing a different computation - for the purpose of the definition is precisely that *the computation should specify the*

functional property and not the other way around. In other words, although it seems to be true that the causal organization of the computer will be different from the coffee-maker's causal organization (hence why one is a computer and the other a coffee-maker), they will still be implementing the same computation so Chalmers' claim should follow, but it doesn't.

Objection 2.

This seems to be the more serious of the two and this is where we encounter the problems with functionalism which I referred to before the beginning of the section. It might be possible to argue that states, such as being a coffee-maker or being Eeyore's birthday present are either simply not functional states of physical systems, or at least not functional states in the same sense as mental states (psychological states at least) are supposed to be functional states of the brain. I shall devote some space to clarifying this problem:

3. Functionalism and Computability

Both of the works Chalmers refers to in connection with psychological properties are classics in the literature of the history of physicalism, which is basically a history of the attempt at the philosophical clarification of the implications of the assumption that the mental is dependent upon the physical. Some of the problems in this area, as opposed to being old hat[25], are still hotly debated even on the global scale, and it appears that there are a wide variety of consistent options available, some considered more plausible by particular theoreticians than others. So called "emergentism" is for instance a different doctrine from central-state identity theory. The first implies the second but not the other way around. Fodor (1982) has a detailed discussion of these options but what is

[25] As the layman thinking himself in possession of a scientific world view would expect them to be.

important for our purposes is his discussion of functionalism. As I had remarked earlier I shall take Fodor's discussion as a paradigm presentation of functionalism in this context, in particular because it will also serve to point out some additional differences between Chalmers' proposed framework from traditional CTM. Here's what Fodor says:

"I remarked that a standard form of materialistic monism is now a functionalism grounded in the machine analogy. [...] The intuition that underlies functionalism is that what determines which kind a mental particular belongs to is its causal role in the mental life of the organism. Functional individuation is individuation in respect of aspects of causal role; for purposes of psychological theory construction, only its causes and effects are to count in determining which kind a mental particular belongs to." (Fodor (1982) p. 11)

Functionalism is a theory which is consistent with physicalism but it is not a logical consequence of it. Fodor's construal of functionalism (and his reasons for adopting it) is informed by two other aspects of his theory, namely that behaviourism is insufficient for accounting for mental life and that the right picture of thought is the representational theory of mind, i.e. the claim that thought is symbolic activity. I have already "discussed" (if such a brief glance could count as a discussion) the second assumption in the previous chapter, so here the following should suffice.

Functionalism is a supporting intuition for the language of thought precisely because it asserts that it is at least possible to construe mental particulars as physical particulars, i.e. as symbols. Functionalism, on the other hand, is a supporting intuition against behaviourism because it *allows for* the construal of mental entities hence avoids the *identification* of mental states with overt behaviour. Mental entities are construed precisely as functional entities. However, functionalism can be argued for on independent grounds, in fact all one needs is an assumption of physicalism and as I said, functionalism will turn out to be consistent with it.

There is, of course, a different question about *what kind* of mental entities are to be

identified functionally. This is an important limitation, so I shall quote Fodor's description of the problem before I say more about what it means for a property to be defined functionally in general.

"[...] we really need a division of the question. I've been running the discussion more or less indifferently on *having a pain* and *believing that P*, but the functionalist story is not in fact equally plausible in its application to qualitative phenomena and to propositional attitudes. It's not hard to see why this is so. Functionalism applies only to kinds whose defining properties are relational. And while it is arguable that what makes a belief - or other propositional attitude - a belief that it is a pattern of (e.g. inferential) relations that it enters into, many philosophers (I am among them) find it hard to believe that it is relational properties that make a sensation a pain rather than an itch, or an after-image a green after-image rather then a red one. What convinces those of us who are convinced is the following (putative fact): Though "qualia inversion" is conceptually possible (my experience might be just like yours, but what looks red to you might nevertheless look green to me), "propositional attitude inversion" is *not* a conceptual possibility. It makes no sense to speak of my belief being different from yours despite the identity of their inferential (etc.) roles. This asymmetry is-plausibly-attributable precisely to the relational character of beliefs; that the belief that P is one from which Q is inferable is, so the story goes, constitutive of its being the belief that P. [...] Since I have nothing to add to this discussion, [...] I shall simply assume that the functionalist program applies at most to the analysis of propositional attitudes." (Fodor (1982) p. 16-17.)

Now Chalmers makes it quite explicit in his paper that he differentiates between psychological properties and phenomenal properties precisely along the same lines as Fodor differentiates between propositional attitudes and qualia. Thus when Chalmers talks about psychological properties, we can at least assume that he includes propositional attitudes among those. An interesting consequence of Chalmers' framework, however, is that he claims that he can demonstrate by a thought experiment, that qualia (i.e.,

phenomenal properties in Chalmers' terminology), such as seeing a particular colour or having a pain, are specifiable computationally - given the assumption that psychological properties are. This consequence does follow in fact, I think, from his assumptions, (see Section 4. where I cite his argument in full) so my task is primarily to demonstrate that being functionalist and physicalist about psychological properties is still insufficient for proving their computational specifiability.

What is important about all this is that functionalism with respect to mental properties is *no more, no less* than what Fodor assumes in the above quote (taken together with some remarks about physicalism which I will not discuss here). Thus in particular when I said before that I don't know how the property of being a coffee-maker is a different functional property from say having a certain belief that P, it turns out, that - at least on Fodor's construal - there is actually *no difference whatsoever*. And I claim that there would be no difference if we examined other descriptions of functionalism in the literature, for there is no difference. As far as functionalism goes, being a coffee- maker is just as much a functional state of a physical system as instantiating a certain propositional attitude. Hence if my little demonstration above proves that the property of being a coffee-maker is not specifiable computationally, then this goes for mental states as well. The insight, that Chalmers' thesis of computational sufficiency has to be false precludes the validity of whatever arguments he presents for the computational specifiability of functional properties. This is true because - if my thought experiment is valid - there are some functional properties which cannot be computational.

In other words, there are two possibilities. The first is that Chalmers has to come up with an argument that shows that mental states are somehow different from other functional states, in which case he will have to produce a different paper in which he accounts for how that difference plays a role in the argument. In the second case, if there is no difference, then there is nothing more to say.

Now in fact it *is* possible to think of some differences, such as the fact that mental states do not seem to have kinds of properties which most physical objects do have. For instance, the property (or state) of being a tea-cup is a functional property, but one obvious reason why it cannot be specifiable computationally is, that having a certain

shape seems to be a necessary property of a tea-cup, and spatial organization is not something which is readily reducible to interactions between states. Mental entities by contrast do not appear to have spatial properties[26]. In fact, according to some theories, the property of being a thought is more or less the same as being a piece of information, and information is in turn an ontological primitive[27]. Whatever one may think about this question, one thing is for sure, that Chalmers never argues for the kind of differences which would facilitate his arguments.

The chief reason which makes it very unlikely that Chalmers could come up with a good argument along these lines is: physicalism. The crux of the argument is precisely that there is a physical system which instantiates mentality (the central nervous system more or less, although there are arguments based on independent grounds, that it makes no sense to speak about mental life without sensory input, i.e. mentality cannot be defined apart from large parts of non-mental systems (see Damasio (1994)). Now, from the fact that mental properties are functional properties of the physical system, it by no means follows that **those properties of the physical system which serve to instantiate the mental properties are themselves, in turn, functional properties of the physical system.** In fact these instantiating properties would have to be such as to be no longer

---

[26] My proof - in this respect - amounts to something like an inverted Chinese-room argument. Thus, if there were a way to reduce the physical system underlying thought computationally and I implemented the corresponding CSA on my computer, I would still have to say that my computer doesn't think. Chalmers could claim however that it thinks, only there is no way of knowing that it does. Thought does not necessarily require expression, it does not have any properties which are accessible to sensory modalities (i.e. you cannot hear, see, touch, taste or smell a thought.) The reason why the Chinese-room is implausible is because it is just too clever. The reason why the implemented computation is implausible is that it is literally too dumb for words. Neither of these, I confess are knock down reasons.

[27] "Some philosophical questions may be so basic that they are never wholly settled. Several scientists have argued that the mind-body problem may be one of these, and that in fact 'mind' has emerged in modern science in the guise of 'information', which plays a central role in physics, biology, and the information sciences. Information is said to be irreducible to physical quantities. As Norbert Wiener has written: >>Information is information; it is neither matter nor energy.<<" (Baars (1988) p. 365.)

susceptible to the argument against computational specifiablity. Thus if Chalmers' scheme is going to work with psychological properties, he will have to show that the relevant properties of the brain are all functional properties of this sort. In what follows I shall refer to these as super-functional properties, and I shall be concerned with them mainly in the next section[28].

It is worthwhile pointing out that Fodor's functionalism assumes nothing of the kind. What makes Fodor's use of functionalism interesting is the extra assumption, namely that functionally individuated mental entities are in fact symbolic in character. Functionalism allows for the hypothesis of mental symbols, but *it doesn't explain them.* In fact in the previous chapter I have emphasized in some detail that CTM is usually not taken to be an explanation of the ontology of intentionality, merely the *character of thought.* It is apparent from the quotes I have given from Fodor, that he allows for the role of mental particulars to be defined in part by reference to their relations to other mental particulars. (E.g. the fact that from a certain belief a different belief doesn't follow serves in part to individuate those beliefs.) There is nothing about Fodor's functionalism (nor in any other version) which requires that all functional properties should be definable in terms of functional (not to mention super-functional) ones. For all Fodor cares, it might turn out that the nature of the physical system underlying mentality is such that it is impossible to realize in other than a biological system. It might turn out otherwise and there is one more point to make about this. I shall then turn to Chalmers' argument about the connection between computation and causality.

It is important to notice that all of the above says nothing about the so called problem

---

[28] Although I take it that all this shows at least the implausibility of Chalmers' scheme, and my task is finished in this respect, I still think it is important to look at how Chalmers argues from functionalism to computational specifiability, for it offers even deeper reasons why this particular attempt at laying a computational foundation for cognitive science is unsuccessful, and it also occasions some thoughts on why it is not very likely that it can be succesful. Besides, functionalism in the philosophy of mind is not the main topic of this chapter and an independent discussion would go along very different lines.

of "multiple realizability". This is an important supporting intuition for functionalism, described in the following quote from Fodor:

"Given the sorts of things we need to say about having pains and believing Ps, it seems to be at best just accidental, and at worst just false, that pains and beliefs are proprietary to creatures like us; if we *wanted* to restrict the domains of our psychological theories to just us, we would have to do so by ad hoc conditions upon their generalizations. Whereas, what does seem to provide a natural domain for psychological theorizing, at least in cognitive psychology, is something like the set of (real and possible) information processing systems. The point being, of course, that there are possible - and for all we know, real - information processing systems which share our psychology (instantiate its generalizations) but do not share our physical organization." (Fodor (1982) p. 9.)

What I would like to point out that all this goes for functional properties such as being a coffee-maker. An Italian espresso machine satisfies the same sorts of generalizations quae coffee-maker as the all-American Drip'o-Matic. In particular, from the arguments I have so far given against Chalmers' scheme, it doesn't follow that thinking cannot be computational. What follows is that even if a computer turns out to be literally a thinking machine, this is not going to happen for the reasons Chalmers advances. However, it is just as important that from the fact (or hypothesis) of multiple realizability, nothing follows with respect to the functional nature of the properties that *realize* the functional property. Thus, multiple realizability cannot be used as an argument in support of Chalmers' claims.

4. Causality and Implementation

Up to now I have basically discussed Chalmers' notion of what it means for a physical system to implement a computation and I have attempted to show on independent grounds that it is not plausible that two systems implementing the same computation will share

their functional properties. Thus, so far I haven't touched upon concepts which are introduced in the second part of the paper, which are supposed to provide positive evidence in support of the claim that functional properties are computational. Of course, if my arguments are correct, then Chalmers' arguments have to be wrong, and in this respect there is no reason to spend time looking at an argument which we know in advance to be fallacious. On the other hand, I believe that the fallacy is an edifying one, not to mention the fact that there could be someone whom my arguments did not convince. Presenting a second argument, then, does not seem to be entirely without purpose.

If the reader will flip back for a moment to where I quoted Chalmers on psychological properties, and ponder the quote carefully, he will notice that the huge transition (which takes place within a single sentence) - which immediately transcends the conclusions of the functionalist position - occurs where Chalmers asserts that "a role within an overall causal system" is the same as "the pattern of interaction between states". In terms of the terminology I have introduced, this counts as a transition from classical functionalism to super-functionalism. This is the key assumption and it essentially involves a hypothesis about the nature of causality. The point is, that even if functionalism could be said to imply the first part, there is nothing to support the second part. To be fair to Chalmers it is important to look at in rather more detail how he argues for this.

Having produced the definition of implementation, Chalmers repeatedly emphasizes the central idea:

"The above account [i.e of implementation P.H.] may look very complex, but the essential idea is very simple: the relation between an implemented computation and an implementing system is one of isomorphism between the formal structure of the former and the *causal structure* of the latter. In this way, we can see that as far as the theory of implementation is concerned, a computation is simply an *abstract specification of causal organization.*" (Chalmers Ibid.)

In other places Chalmers emphasizes that the definition of computation should be purely formal, independent of semantics and independent of any 'loaded' notion of syntax. As we have seen Mellor and others are loath to admit that a random Turing machine program could be called a computation. Chalmers rejects this view, and rightly in my opinion. Computation is a mathematical notion (to what degree exactly is to be discussed in the next chapter) and any function is a "meaningful" mathematical object.

It is essentially in this sense that Chalmers' project actually manages to be an attempt at showing what a *foundation of cognitive science should look like*: the notion of computation he works with is as basic as it can get. Concepts like syntax, semantics, information and the like, dispute about whose nature we have seen to be central in Mellor's paper, play no role whatsoever[29]. On the other hand they are replaced with a notion that is even more obscure in my opinion, namely causality.

The outline of the argument is as follows:

"Causal organization is the nexus between computation and cognition. If cognitive systems have their mental properties in virtue of their causal organization, and if that causal organization can be specified computationally, then the thesis of computational sufficiency is established." (Chalmers Ibid.)

There are two statements whose truth needs establishing. The first is that "cognitive systems have their mental properties in virtue of their causal organization" and the second is that "causal organization can be specified computationally". Chalmers takes the truth of the first statement, as we have seen, to be given by the arguments for functionalism. This is a rather tricky point, because in some sense it is true. Whatever it is that physicalism asserts, it will assert something like the mental being dependent on

---

[29] "I have said that the notion of computation should not be dependent on that of semantic content; neither do I think that the latter notion should be dependent on the former. Rather, both computation and content should be dependent on the common notion of *causation*. (Chalmers Ibid.)

the physical, i.e., on the "organization of matter". Saying "causal organization" instead of "the organization of matter" seems to be just employing common parlance. And in fact, in the paper this is precisely what Chalmers does. He asserts (without an argument) that psychological properties are dependent on causal organization. He also asserts something about "causal organization" which allows him to show that it is specifiable computationally. It then turns out that the dependance of psychological properties on causal organization is tied to the same fact about causal organization which is assumed by computational specifiability. The part that never gets proved is precisely the "transition" I referred at the beginning of this section.

First, let's look at the argument for the computational specifiability of causal organization. I shall quote Chalmers in full on this point.

"To spell out this story in more detail I will introduce the notion of the 'causal topology' of a system. The causal topology represents the abstract causal organization of the system: that is, the pattern of interaction among parts of the system, abstracted away from the make-up of individual parts and from the way the causal connections are implemented. Causal topology can be thought of as a dynamic topology analogous to the static topology of a graph or a network. Any system will have causal topology at a number of different levels. For the cognitive systems with which we will be concerned, the relevant level of causal topology will be a level fine enough to determine the causation of behaviour. For the brain, this is probably the neural level or higher, depending on just how the brain's cognitive mechanisms function.

Call a property P an organizational invariant if it is invariant with respect to causal topology: that is, if any change to the system that preserves the causal topology preserves P. The sort of changes in question include: (a) moving the system in space; (b) stretching, distorting, expanding and contracting the system; (c) replacing sufficiently small parts of the system with parts that perform the same local function (e.g. replacing a neuron with a silicon chip with the same input/output properties); (d) replacing the causal links between parts of the system with other links that preserve the same pattern of dependencies (e.g. we might replace a mechanical link in a telephone exchange with

an electrical link); and (e) any other changes that do not alter the pattern of causal interaction among parts of the system.

Most properties are not organizational invariants. The property of flying is not, for instance: we can move an airplane to the ground while preserving its causal topology, and it will no longer be flying. Digestion is not: if we gradually replace the parts involved in digestion while preserving causal patterns, after a while it will no longer be an instance of digestion: no food groups will be broken down, no energy will be extracted and so on. The property of being a tube of toothpaste is not an organizational invariant: if we deform the tube into a sphere, or replace the toothpaste by peanut butter while preserving causal topology, we no longer have a tube of toothpaste." (Chalmers Ibid.)

The first thing to notice about causal topology is that it doesn't exist. Chalmers asserts that it exists without arguing for it (in fact the way it is described I have a hard time seeing how one could argue for it) and it is not hard to see why. Chalmers would simply like to avoid a circular argument, i.e., just stating out of the blue that causal organization is specifiable computationally. The way he is going to avoid this circularity is by interposing the notion of causal topology. Causal topology, in turn, is nothing else but the implementation of a computation described in a somewhat different language and called by a different name. That it turns out to be computationally specifiable is no surprise.

One of the reasons why it would be difficult for me to argue this conclusively, is that the way Chalmers sets it up the counter-arguments would have to involve going much deeper into notions like causality than he in fact does. Nonetheless I will point out some problems with the scheme.

First of all, Chalmers does everything to obscure the fact that a system cannot be said to have causal topology except *relative to a certain organizational invariant*. This, I take it, is an obvious point. The first move toward obscuring this is the introduction of the notion of "causal topology" before "organizational invariance". Chalmers then talks as if "organizational invariants" were invariant with respect to a *given* "causal topology".

He also talks about some properties not being "organizational invariants".

This last claim is simply false. All properties are "organizational invariants" precisely with respect to the level of "causal topology" that *they specify*. Chalmers says that the property of being a tube of toothpaste is not an organizational invariant because if we change the paste to peanut butter *while preserving causal topology* we will have thereby destroyed the property of being a tube of toothpaste. The point is precisely that the level of specification of the property of being a tube of toothpaste *includes* containing toothpaste, this belongs to the "causal topology" underlying the property of being a tube of toothpaste.

The intuition behind Chalmers' idea is a plain one. It seems to be a metaphysical truth that given a physical system P, and some property p of that system, one will always be able to find other properties of P with respect to which p will be invariant. The fact that people can have artificial hearts is a case in point. The property of being alive is invariant with respect to the property of one's heart being made of flesh. All that a claim like this would involve, however is the familiar hypothesis of "mutiple realizability" and as we can expect, Chalmers is aiming for something much stronger.

The way around this predicament is hidden in the first paragraph of the above quote where it says "abstracted away from the make-up of individual parts". In other words the extra restriction which Chalmers wants to put on the notion is precisely one which will only allow for functional properties to be organizational invariants, what is more not just functional properties in the sense of functionalism, but the kind of functional properties which do not depend on properties which cannot be specified computationally. The problem is that there is no way to describe properties like that except in this very way, as ones which can be specified computationally. These are precisely the kind of properties I referred to as super-functional.

Perhaps I shouldn't stress the obvious. The following quote, in which Chalmers wraps up his argument suffices to reinforce what I have been saying.

"An organizational invariant property depends only on some pattern of causal interaction between parts of the system. Given such a pattern we can straightforwardly

abstract it into a CSA description: the parts of the system will correspond to elements of the CSA state-vector, and the patterns of interaction will be expressed in the state-transition rules. This will work straightforwardly as long as each part has only a finite number of states that are relevant to the causal dependencies between parts, which is likely to be the case in any biological system whose functions cannot realistically depend on infinite precision. Any system that implements this CSA will share the causal topology of the original system. In fact, it turns out that the CSA formalism provides a perfect formalization of the notion of causal topology. A CSA description specifies a division of the system into parts, a space of states for each part, and a pattern of interaction between these states. This is precisely what constitutes a causal topology." (Chalmers Ibid.)

There are two mysteries remaining. The first is: why does Chalmers think that psychological properties are totally independent of the make-up of the brain; the second is: how does the concept of causality enter the picture.

To answer the first question it will be enough to look at the proof Chalmers gives of phenomenal properties being "organizational invariants". In section 3., I quoted Fodor who expressed what seems to be a general concensus among the majority of philosopers thinking about these matters, namely that phenomenal properties (such as seeing a particular color) do not seem to be functional properties. According to Chalmers they're not just functional but super-functional. Why? To put it bluntly, Chalmers thinks that the brain is made up of things called neurons which are like switches which can be on or off. Whether they are made of plastic or proteins is beside the point. He also happens to think that *there are no properties of the brain which cannot be reduced to properties whose raison d'être is to support one or another super-functional state*. Now, there is absolutely no evidence for this, in fact much of the popular neuroscience literature is about how various chemical properties and even the shape of the cells is causally relevant for mentality. I personally don't know much about neuroscience, but I know enough to realize that the picture Chalmers sports of the brain is somewhat medieval; it was

outdated even when it was advanced, if it ever was advanced in this form[30].

This is not to say of course that when all the cards are on the table, Chalmers could not turn out to be right. The likelihood of this is however very, very small. Also, this is why I went to some pains to emphasize that "multiple realizability" is not evidence for his claims, for it isn't. Each of the realizations could turn out to be dependent on non-super-functional properties.

Anyway here is Chalmers' "proof" which is also the last part I shall quote:

"The argument for this, very briefly, is a *reductio*. Assume conscious experience is not organizationally invariant. Then there exist systems with the same causal topology but different conscious experiences. Let us say this is because the sytems are made of different materials, such as neurons and silicon; a similar argument can be given for other sorts of differences. As the two systems have the same causal topology, we can (in principle) transform the first system into the second by making only gradual changes, such as by replacing neurons one at a time with I/O equivalent silicon chips, where the overall pattern of interaction remains the same throughout. Along the spectrum of intermediate systems, there must be two systems between which we replaced less than ten percent of the system, but whose conscious experiences differ. Consider these two systems, **N** and **S**, which are identical except in that some circuit in one is neural and in the other is silicon.

The key step in the thought-experiment is to take the relevant neural circuit in **N** and install alongside it a causally isomorphic silicon back-up circuit, with a switch between the two circuits. What happens when we flip the switch? By hypothesis, the systems conscious experiences will change: say for purposes of illustration, from a bright red

---

[30] For an old a priori argument why this approach is barbaric see von Neumann (1958) where it is argued that if simply assume that neurons are parts of a digital computer then the figures won't work out even if we suppose that it is massively parallel. Additional properties seem to be involved. For some data on neuroscience see Churchland-Sejnowski (1992). On the other hand there are independent ontological arguments which make it very unlikely that anything remotely resembling Chalmers picture could be true (See Martin (1993)).

experience to a bright blue experience (or to a faded red experience, or whatever). This follows from the fact that the system after the change is a version of **S**, whereas before the change it is just **N**.

But given the assumptions, there is no way for the system to notice these changes. Its causal topology stays constant, so that all of its functional states and behavioural dispositions stay fixed. If noticing is defined functionally (as it should be), then there is no room for any noticing to take place, and if it is not, any noticing here would be a strange event indeed. There is certainly no room for a thought >>Hmm! Something strange just happened!<<, unless it is floating free in some Cartesian realm. [...] This, I take it, is a *reductio ad absurdum* of the original hypothesis: if one's experiences change, one can potentially notice in a way that makes some causal difference." (Chalmers Ibid.)

For all its neatness I think this argument mainly serves to underline the kind of problems I have been talking about. Besides all the problems I have mentioned, Chalmers also dismisses in a few curt remarks the whole debate about the role of the environment in determining mental content. On some construals (e.g. Putnam (1988)) this alone is sufficient for rejecting functionalism. I also feel that speculation about how the thesis of computational explanation would follow even if I could accept the thesis of computational sufficiency would be beside the point[31].

About the other "mystery": One way in which one could roughly reconstruct Chalmers' argument is the following:

---

[31] Even if it were true, Chalmers would need to argue that separate sub-circuits could easily be identified for particular mental processes and they will be sufficiently simple (and structured) as to yield insight into the mechanisms. This is by no means a trivial consequence of the foregoing (see e.g. Fodor (1987)). I have not mentioned - because earlier I said I wouldn't get into the topic - that Chalmers' picture of the mind is essentially the so called "connectionist" model. No serious connectionist however would go as far as Chalmers. See Clark (1989) or Churchland-Sejnowski (1992) for discussions about how connectionism is an intuitive research tool.

a, Mental processes are material processes;

b, Material processes are causal processes;

c, Causal processes are "patterns of interaction between states".

The transition between b and c is what Chalmers employs in shifting from functionality to super-functionality and in going from causal organization to computationality. This argument - with which Chalmers would perhaps agree with in this form, perhaps not - is one of the things which is behind this type of thinking and it is based, I believe, on a seeming overlap between the theory of physical determinism - which is essentially about causality - and the theory of algorithms. This is a raw intuition of mine; manifest mostly on the level of musings and ruminations. What made me think about it, however, is the fact - which I was hoping would emerge from the last two chapters - that there is something obviously wrong about computational theories of mind, but there is also something obviously wrong about arguments advanced against them. My intuition is that there must be an equivocation somewhere at the level of our most basic concepts. The closing chapter is an attempt at saying a few words about where this might be taking place.

Chapter IV.

Transcendental Meditations

In the last two chapters I have examined two proposals in detail on how to settle once and for all the question whether the mind is computational. In the introduction to the essay I mentioned that there would be some common features of the proposals, the most prominent of which was that both authors considered the interpretation of the term 'computational' to be problematic, and hence found it necessary to inquire into its true meaning. It appears - after taking a look at the arguments - that what looms large behind both definitions of computation is the *mathematical* theory of computations, the formulation of which goes back to Turing, but neither philosopher seems content with what Turing's formulation has to offer.

In what follows I shall argue that the main problem with their arguments is, that although they reject a purely mathematical theory of computations, both Mellor and Chalmers (and other theoreticians of the sort) hold on to the idea that in some way or another it will be *algorithms* which will have to provide the link between the mental and the material, or in other words, the theory of physicalism will have to be cashed out in terms of the theory of algorithms. On the other hand, there is no alternative theory of algorithms besides the mathematical theory.

In order for me to give this argument in more detail I will have to talk about these concepts at some length. I shall begin by saying what *I think* the term computational covers, then I will say a few things about algorithms. Along the way I shall be pooling these digressions for intuitions that allow for some "transcendental speculations" as to what it is about these concepts that brings about the controversy.

1. Computations according to me

When we talk about computers, the most important fact we have to keep in mind - I think - is, that they consist essentially of two parts. A computer is a machine which can

calculate a certain class of fuctions defined on the integers (such as addition or multiplication) - the so called effectively calculable functions. This is essentially the "core" of the machine in a sense. The other part consists of a "representational system" of more or less elaborate complexity which is the part that is actually "useful" to us, i.e., which turns the device into a "machine". In the case of the everyday computer this division is manifest roughly between the terminal and the processing system. Here are a few examples of how the division works in practice.

1. The word-processor I am using to write this text is a good example. The computer can do all sorts of things with the letters and the words and sentences which they form: display them on the screen, store them, move parts of them around, check their spelling, etc. This is partly due to the representational part, i.e., that the computer is equipped with a TV screen and a keyboard. On the other hand it also has some unseen internal mechanisms in virtue of which it can execute certain operations on these "representations". More precisely, it executes operations on "representations" of these "representations", where the former use of the term refers to the fact that the internal operations of the machine are not defined over things such as letters on the screen. I will say more about these two uses of the word "representation" below.

2. In a video-game, say in a car-racing game, the "representational system" becomes less "innocent". Thus it will have to be good enough to generate some excitement in the user about the game. The drawings displayed on the screen will have to be realistic in ways that allow some voluntary illusions to arise in the player about taking part in the real thing. However, even in the case of the video-game, it is still a clear case of *simulation*. The representational system and the machine which makes the drawings move around are clearly separable, due especially to the fact, that we know that this is the same sort of computer that makes our word-processor work[32].

---

[32] This is not the same as the so called problem of virtual reality. In fact I don't think that that is a problem, because I don't think that virtual reality is possible. Briefly, there

3. The division begins to get less clear cut when instead of a "representational system" and the computing part we have some physical system *controlled by a computing system.* In this case the "representations" the computing system works with will no longer be "representations" meaningful to us, but "representations" meaningful to the system itself, whereas what is meaningful to us will not be "representations" in the literal sense, but something intended to be the *real thing*. This is less complicated than it sounds on the one hand, but it is probably what gives the main aspiration to AI workers. Robots are no longer representations governed by computations but they actually do things. When a robot moves its arm that is not a simulation of moving an arm it is actually the movement of an arm. On the other hand it is what I am calling the "representational system" that has changed, the computing part, which controls the movements of the arms, remains the same.

It is perhaps not so clear what the difference is between the two parts after all. The reason behind this is that the two parts can be more or less involved with each other, and in some cases the prominence of one part could be overwhelming. A handheld calculator for instance is no good for anything else besides calculating numerical functions. It's "representational system" is very limited. Also, because it doesn't usually have a memory where it can store programs (that is a series of instructions for which operations to perform on an arbitrary input) even the class of functions it can compute seems very limited (although it can compute any function in an interactive way). If we consider game-playing machines such as a chess-machine or even a gambling machine, the computational part of these seem insignificant compared to the "representational system".

---

could be two kinds of things which could claim the status of VR. There could be some world created inside the machine which feels real and somehow we are able to manipulate it. I take it, that we don't even know what this would look like. The other kind would be the one with the evil neuroscientist putting our brains in a tank and stimulating our nerve-endings to make us believe that various things are happening to us. This has the reality of a thought experiment but one of the problems is, that I don't know how one could argue that this would be *virtual* reality subjectively and not the actual thing. What's the difference?

The computing part of the machine is thoroughly specialized to manipulating the "representational system".

Looking at computers from a purely phenomenological standpoint, this distinction is much more important in my opinion than the one which is usually at the forefront, namely the hardware/software distinction. The latter emphasizes that the difference between a computer and an average machine is that the former can be programmed for different tasks - it is actually an infinite medley of machines built into one. What *I* would like to point out is, that this character of the computer would be quite insignificant, were it not for the purely technological innovativeness of the representational system to which it could be hooked up. From the three examples I have given above it should be clear what I mean when I say the "phenomenology of computers". It is precisely how the inputs and outputs of computations appear to us after going through their metamorphosis in whatever representational system the processors are hooked up to, that is relevant in categorizing the machine (for instance quae simulator or instantiator of certain properties).

For the sake of clarity, from now on I shall always call one part of the computer the "representational system" (or RS for short) and the other part the "computing system" (or CS) for short (and I will drop the quotation marks). It is my opinion that both Mellor and Chalmers characterize *computational physical systems* in the wrong way. Mellor's mistake in defining computation is the dogmatic insistence that the computation has to be the processing of propositions in terms of the processing of physical tokens of the propositions. According to this definition, neither a chess-machine nor any video-game (nor any robot such as an industrial one) is a computational device. The problem with Chalmers' definition is that first of all it is not at all clear that it is technically feasible to define implementation in his terms, and compared with my scheme (which I am about to develop) his fails to differentiate between the CS and the RS parts of computers. Perhaps this is what leads him into describing causal processes mistakenly as computations with unwarranted generality (although I will have more to say about this).

I haven't yet said how *I* would define a computation, but I happen to think that precisely because of the differentiation I have suggested, it is perhaps quite impossible to give a definiton which will manage to avoid ambiguous cases *in toto*. The part which one can define unambiguously is the CS part. This is the part for which a variety of mathematical formalisms are available, the most popular of which are Turing-machines. The reason why they are the most popular is because they are enlightening precisely with respect to the connections and differences existing between CS and RS.

I gave a brief definition of Turing-machines in connection with the Turing-test in the introduction, but for what I want to say about them now it won't hurt if the reader keeps in mind FSAs - introduced in chapter two - as well. The interesting thing about these formalisms is that they are not formalisms for calculating functions on the integers to begin with; they do so only in an interpreted sense. One needs to interpose some sort of *coding system* in order to see how these formalisms define functions on numbers[33]. In the case of Turing-machines, the kind of entities a Turing-machine works on are strings of symbols, or in other words "formal languages". There is no space here for an excessive description of what a formal language is, so if the few sentences I will say about them seem vague, the reader is advised to look them up somewhere. Turing-machines are devices which are apt at handling tasks having to do with the structure of the strings of symbols and are also *mechanical* (or algorithmic). The point about Turing-machines is that they seem to be the most general kind of devices for handling these kinds of tasks. If a Turing-machine cannot do it, then no other machine can[34]. Now, the

---

[33] A formalism defined explicitly for the effectively calculable functions on the integers is Church's Lambda calculus. A popularized description of this system can be found in Penrose (1989).

[34] Tasks like these, for instance, are "calculating" functions from sets of strings to sets of strings. A special case would be "recognizing" strings which belong to a particular language (a language being a subset of the possible strings which can be formed from the alphabet of symbols). One way to implement this is to have the machine write one kind of symbol as output if the string belongs to the "language" and a different kind of symbol if it doesn't. This would amount to computing the so called "characteristic function" of the language.

point is that a Turing-machine itself is a kind of machine which is essentially a mix between the CS and the RS parts - indeed, it is precisely a formalization of how these two come to be mixed. If we apply a suitable coding system we will get a numerical function. (For instance if we code the set of symbols in binary (0 and 1) and take care that inputs and outputs should not begin by 0, then we get the CS part of the machine. This is because these binary codes have an interpretation as numbers for all functions that the computer calculates on the strings, i.e., on the representational system.) And more importantly, the fact that the Turing-machine calculates this numerical function is totally independent of the fact that it also happens to calculate a certain function on representations where those representations could be meaningful (i.e., answers to questions *or* proofs of theorems *or* events of a simulated car race *or* electrical pulses which serve to initiate certain mechanisms which in turn realize phases of the movement of a robot's arm).

The point I intended to make originally was about the non-ambiguity of the CS part of computers. By this I meant that the CS part will be the part which - under a "suitable coding system" - will be found to be calculating one of the effectively computable functions. The Turing-machine is one of the formalisms which can calculate all of these and only these functions. The bit about the "suitable coding system" is in turn what makes the definition of RS problematic. It would perhaps be more appropriate if instead of suitable coding system I were to say "*suitable coding systems which code the proximal input to CS from RS*". However, in real life cases this is precisely the part which becomes impossibly vague. First of all, most of the time there are several levels of "representations". The representations which we use to interact with the machine, and in whose terms the machine will actually be useful to us, might be many times removed from the kind of objects on which basic operations are defined within the "CS unit", and these are the kinds of objects in terms of which we want to interpret the CS part as calculating one of the integer functions[35].

---

[35] I will not be getting into the problem of analog vs. digital machines here. There is a controversy in the literature about how an analog machine differs from a digital one,

And the situation is even worse, because a clearly identifiable CS part might be missing altogether. Many systems (such as my household machines in chapter 2.) can be interpreted as input-output systems, yet they will have nothing like a CS system which calculates outputs from inputs. It is essentially these cases where one is reluctant to call what is a happening a "computation" although there are other - borderline - cases where there is a specialized part actually controlling what is happening in the machine as a whole (e.g. a fuse in any electronic device). But then, in the case of household machines, even the RS part could be said to be missing[36].

My view is that it is probably impossible to give an exact definition of a computational physical system, and the reason for this is, that **we don't know what kind of representational systems can be hooked up to computing devices and in what ways,** because this is not a matter of a priori theorizing, but of empirical discoveries. It is also a matter of discovery which natural processes will turn out to be computational even in this sense. One of the transcendental reasons for the opposition between Chalmers and Mellor, is thus, that there is an indefiniteness here. Computation enthusiasts like Chalmers tend to overinterpret the lack of a definite boundary between CS and RS, while bitter opponents of the outlook - like Mellor - tend to overemphasize the clarity of the

---

and what a hypothesis which says that the brain might be an analog computer, actually means. (See Pylyshyn (1984) and von Neumann (1958). It seems to me that adopting my terminology, one could try to categorize computations into analog and digital (in a way useful for cognitive scientists) by the complexity of the levels of representation-transformations.

[36] One might think of Mellor's example of the absurdity of the mass calculating the resultant vector of the velocities, as typically a system which lacks the CS part. Mellor's definition is much more restrictive than this, however. There could be cases, where I would say that there is clear cut evidence for the presence of CS systems, which would still not be processing of propositions. In chapter one I mentioned that Mellor dismisses the possibility of a system's being computational (syntactic, to be more precise) in virtue of implementing an algorithm. My framework together with what I will say about algorithms is in part a vindication of this possibility. The tricky part is to realize that the "representations" that the algorithm is enacted upon are no longer meaningful ones, and that a procedure's being a Turing machine computation is totally independent from it being "semantic" in any sense.

difference between the two features.

Here, then, is a summary statement (instead of a definition) of what I think a computational system is.

What I would like to call a *semi-computational system* is an input/output system which possesses subsystems which "actually compute" in the sense that their inputs and outputs can be interpreted under a suitable representational system as inputs and outputs to Turing-machines, and their role in the workings of the system as a whole is defined solely in terms of these inputs and outputs. I would call a system *computational* instead of semi- computational if this division into CS subsystem(s) and RS systems is *fairly clear cut* and the complexity of the CS system is quite elaborate (i.e., for instance responsive to multiply structured inputs composed of some "alphabet"), such as is the case with the digital computer. In the case of the fuse box, the problem is with the impoverished complexity of the input to the CS system, in the case of the brain one of the problems is with the clear-cut part[37].

One important feature of this "definition" (at which I have hinted already in example 3.) is that the role of the term "representation" has shifted from the role it usually plays in definitions of computations. Mellor, for instance, did not want to call a system computational unless what it processed were representations, the latter phrase meaning tokens of things which could be broadly classified under the heading of "meaningful to

---

[37] An extra qualification one could put on the CS system of a computational system is that it should be "programmable". This condition seems to be too strong however (as I mentioned before), so one could say instead that it should be the kind of structure of which different versions do slightly different things precisely as if they were running different programs. In other words the CS system should be some sort of instantiation of a hard-wired function, such that another CS system could be realized on similar architectural principles instantiating a different "computation". It is not clear for instance that the mechanisms in a fuse box are even remotely like this. Whereas the complexity condition would make the difference between a fuse box and a computer merely quantitative, this makes it qualitative.

us". In my definition, the term representation[38] refers to tokens "meaningful" *to the subsystem*, where meaningful means that it can "recognize" it and respond to it as a Turing-machine does to a symbol.

So far so good. All this however does not really get us that much further. Although I tried to emphasize that the mathematical notion of a computation is quite clear, I also acknowledged that defining computation solely in terms of functions on integers would be an undue limitation on the notion. On the other hand it seems that allowing for a more general picture, involving interacting structures of different nature, blurs the edges of the concept, and now it seems that saying which physical systems are computational is a matter of empirical discovery. We don't know in advance what kind of RSs can be hooked up to CSs but we also don't know a priori how CSs can be realized. But perhaps we shouldn't have to go that far in relativizing the concept of a Computational System. The most important part of my "definition" is - probably - where it says that there should be an identifiable "CS-unit" whose doings should not be relevant to the workings of the system as a whole except insofar as they can be interpretable as Turing-machine computations. In other words, the CS should be the physical realization of a Turing-machine which is the same as saying that it should be executing an algorithm. The notion of "algorithm' is also a transcendental concept in this context, it is one which breeds ambiguity. I shall finish this chapter by saying a few words about this.

2. On algorithms

So far I have advanced the claim that one transcendental reason for the controversy

---

[38] I find it difficult to substitute a different English term here for 'representation' which could do the work. The Kantian use of 'Vorstellung' of which 'representation' is the standard translation is more or less what I am aiming for. In Kant, representations do not necessarily "represent" anything. They stand for whatever could be "present to the mind" - any candidate for being a content of consciousness. In this sense are inputs to the Turing-machine "present" to the Turing-machine, a slight anthropomorphization.

about CTM is, that - contrary to expectations - it is not at all clear which physical systems are computational, because even though computing in the mathematical sense is quite clearly defined, the ways in which a computing system (in the strict sense) can interact with the physical world is an empirical question. Besides this, I think there is at least one more reason (of the kind that I can see), which is connected to what I think are false expectations about how the connection between the mental and the physical *should* turn out to be like.

In this context, the following quote from Daniel Dennett's paper on computational approaches[39] should be interesting because his project is parallel to mine, but I think I have managed to draw somewhat different conclusions. Here's how Dennett describes the transcendental reasons for the prominence of computationalism

"There are still dualists and mystics in the world who assert (and hope and pray, apparently) that the mind will forever elude science, but they are off the map for me. A goal that unites all participants in the conflict area I will explore is the explanation of the aboutness or intentionality of mental events in terms of systems or organizations of what in the end must be brain processes. That is, I take it as agreed by all parties to the discussion that what we want, in the end, is a materialistic theory of the mind as the brain. Our departure point is the mind, meaning roughly the set of phenomena characterized in the everyday terms of >>folk psychology<< as *thinking about* this and that, *having beliefs about* this and that, *perceiving* this and that, and so forth. Our destination is the brain, meaning roughly the set of cerebral phenomena characterized in the *non*intentional, *non*symbolic, *non*-information-theoretic terms of neuroanatomy and neurophysiology. Or we can switch destination with departure and construe the task as building from what is known of the plumbing and the electro-chemistry of the brain toward a theory that can explain - or explain away - the phenomena celebrated in folk-psychology. There has been a surfeit of debate on the strategic question of which

---

[39] Daniel Dennett: "A Logical Geography of Computational Approaches". In: Brand and Harnish (1986).

direction of travel is superior...A much more interesting clash concerns what to look for in the way of interstitial theory. It is here that manifestos about <<computation>> vie with each other..." (In: Brand and Harnish (1986) p. 61.)

I think that Dennett perceives the role of CTM very clearly, for it is indeed an interstitial theory. The part I slightly disagree with is that he overconcreticizes the problem. I don't think that CTM is introduced *as* an interstitial theory about how to reach the available data about neurophysiology and neuroanatomy from psychology or the other way around. As I see it, it is precisely the gloominess of the outlook about whether we will ever be able to accomplish either of these that is behind CTM. The problem is precisely that the kind of things we know about the brain and the kind of things we "know" about mental phenomena just don't seem to fit together. Far from either bundle of facts explaining the other bundle, *we don't even know how to begin constructing a theory which connects the two*. The difficulty of switching ontological planes seems to be slightly reduced in the case of mental phenomena though, for, as we have seen, some computational theory at least seems to provide a framework of explanation of mentality: Fodor's language of thought theory. But as we have also seen, there is not even a hint as to how to move from the symbols toward the brain, and if we can believe Fodor about the appropriateness of quoting Lyndon Johnson ("I'm the only President you've got!") then the prospects of moving from mentality toward neurology are looking poor indeed (since we would need to start with a different framework, but what Fodor's quote implies is precisely, that there is no other framework).

To cut a long story short: I don't think that CTM's assumptions rest on intuitions derived from *concrete* phenomena; the reason for its persistence is much more general.

Instead of the picture Dennett gives I propose a slightly more complicated one. I think the CTM is an interstitial theory not between neurological data and data on mental life, rather it comes to be formulated originally as a plausible hypothesis about the connection between the mental and the physical in general. And even as a theory of this sort it is at first not a computational theory. In other words, though it is true that the problem on the

one hand is connecting what we know about the brain with what we know about the mind, nonetheless CTM interposes a concept which seems to be the right one because of a kind of picture we have of *material processes* and the kind of picture we have of *mental processes*. This concept is that of an *algorithm* and why it seems to be the right one I will explain momentarily. Once this concept is interposed, it then happens that various theories about whether CTM is right or wrong argue by trying to tie algorithmicity to *what we know about the brain and the mind.* Some of the confusion arises because it looks as if the two problems were the same but they are not. Before I say more about this I shall first try saying how I think algorithms come between matter and mind.

3. Algorithms as the "bridge" between mind and matter

One reason (which I haven't mentioned) why the Chinese-room experiment is not very convincing, is that Searle seems in part to be arguing against himself. He admits (see Searle (1992) and also some remarks in Pagels (1984)) that mental phenomena have to have a material basis, and what will count as an explanation of mental phenomena is the reduction of intelligent processes to non-intelligent ones. We need a description of mental processes which can make do without the so called "homuncular" picture Fodor describes:

"Here is the way we tie our shoes:

There is a little man who lives in one's head. The little man keeps a library. When one acts upon the intention to tie one's shoes, the little man fetches down a volume entitled *Tying One's Shoes.* The volume says such things as: "Take the left free end of the shoelace in the left hand. Cross the left free end of the shoelace over the right free end of the shoelace..., etc.

When the little man reads the instruction 'take the left free end of the shoelace in the left hand', he pushes a button on a control panel. The button is marked 'take the left free end of the shoelace in the left hand'. When depressed, it activates a series of wheels,

cogs, levers, and hydraulic mechanisms. As a causal consequence of the functioning of these mechanisms, one's left hand comes to seize the appropriate end of the shoelace. Similarly, *mutatis mutandis*, for the rest of the instructions.

The instructions end with the word 'end'. When the little man reads the word 'end', he returns the book of instructions to his library.

That is the way we tie our shoes." (Fodor (1982) p. 63-64.)

Now, although psychological theories about some aspects of our mental life could manage to get by with something like this story (for instance reducing rationality to a number of deliberation processes classified into types, but still requiring intelligence, like the ones Mellor mentions at the end of his paper) an explanation of mind in terms of material processes is precisely an *attempt at eliminating* these circularities. Searle's arguments against the Chinese-room go against himself, because if the elimination of homuncularities is successful, then we will necessarily end up with something like the Chinese-room. A picture of something inside the brain which (who) actually understands the symbols is precisely the kind of thing Fodor ridicules. On the other hand it seems that this little homunculus is exactly what Searle misses about the room, but he still thinks (of course) that there will have to be some mechanism which does the understanding. This mechanism will in turn be another Chinese-room. Yet, the part which is I think must be close to the truth about Searle's argument is that it won't be exactly like it. What I am getting at is that what we will end up with are not necessarily *algorithms*, but rather: different kinds of physical processes. What Searle's argument points out is that algorithms (programs in his version) do not seem to be the right sort of things in terms of which to explain mentality. But the reasons he gives are not the right reasons.

The right reasons will have to emerge somehow from a different demonstration of why algorithms are inappropriate for bridging the explanatory gap that exists between the mental and the physical.

As Dennett has remarked, there are two kinds of strategies (or at least there used to be) one could employ when looking for the connection between the mind and the brain,

the bottom-up and the top-down approach. Similarly, there are two kinds of strategies one could employ for trying to find the connection between the mental and the physical and these two kinds of approaches amount to two different attempts at eliminating the homunculi. The reason why algorithms appear to be able to bridge the gap is that they are useful *conceptual tools* starting from either direction. However, I think they are useful for different reasons, and in the one case (i.e. in the bottom-up case) I don't even think that their use is as justified as in the top-down case.

When trying to eliminate the homunculi from the mental, it seems that algorithms are the right tool to use, because in a sense this is exactly what they were invented for. What I say about this will necessarily be very cursory.

The name 'algorithm' (we all know that it is of Arabic origin deriving from the name of a corresponding Arab individual) was originally employed for denoting "mechanical" procedures in mathematics. Such a procedure is the one we use to add numbers on paper. The point of the existence of the procedure is precisely that one can *use it without thinking* every time one encounters the problem of having to add two numbers. Similar, well known procedures exist for multiplication and division as well. The example one usually finds cited in introductory discussions of algorithms - as a mathematical method - which is not so well known is the so called Euclidean algorithm, a procedure for finding the greatest common divisor of two positive integers (see: Penrose (1989)). The reason why it is interesting is that it is not so clear that the algorithm actually works, i.e. one has to think for a bit in order to be convinced that what the method yields is the greatest common divisor and that it always yields something, i.e., that the procedure terminates.

It is not so clear which are the kinds of tasks for which algorithms exist and even if in a particular instance it is clear that an algorithm should exist, it may sometimes be extremely difficult to find one. Moreover, it is usually not entirely indifferent *what kind of algorithm* one finds. One of the central and currently open problems of finite mathematics is the so called P = NP problem. P and NP both denote classes of

decision-problems (i.e. problems whose solution consists in a yes or no answer, e.g. whether certain structures contain certain substructures or not). For problems in the P class we have algorithms which in a certain sense are "fast" (i.e. take only a "small" number of steps to execute measured in the size of the "input") whereas for problems in the NP class no such "fast" algorithms are known. The open question is whether the two classes are in fact different or not. The reason why this is interesting for us is, that it shows *that there is a great deal that we don't know about even relatively simple properties of algorithms* and this, I think is an intuitive support for thinking that algorithms may be the right model for mental processes. Although they are simple in a way, their nature is by no means known to us entirely.

So far I have mentioned algorithms only for kinds of mathematical questions which require actual calculations, i.e, operations on numbers. Turing's framework however was invented in the context of a more general question which the mathematician David Hilbert asked, namely, whether there could in principle exist a "mechanical procedure" for deciding of an arbitrary proposition of mathematics if it is true or false. Turing formulated the notion of Turing-machines as a formalization of "mechanical procedure" (or algorithm). He then proved that there cannot be a Turing machine that can decide of an arbitrary Turing-machine computation whether it will ever terminate or not (the so called "halting problem"), and in this sense managed to answer Hilbert's question in the negative. Turing-machines are general because - as I've said before - they are devices for manipulating arbitrary symbol systems, not just numbers, hence they provide a general model for any deterministic procedure described in some representational system.

Now, one has to say that what humans do (i.e. behaviour in general) - if one really is committed to elminating homunculi - is precisely: execute algorithms. Although at first glance the difference between algorithms and "minded" behaviour seems to be precisely that algorithmic is the part where one doesn't necessarily need to be "present", it seems that if we look long enough we will always have to be able to reduce what we do to algorithms otherwise we end up with the little people in our heads tying our shoes. Algorithms are the right model for mental processes <u>not</u> because of what Fodor says, that

what is in our head are symbols; not because of what Chalmers says, that our brain implements a computation. Algorithms are the right model, because it looks as if there were no other model. Notice that although what compels us to be sure that homoncularities should be eliminated is that the physical world is deterministic and the mental by assumption (see Dennett in the quote above) is part of the physical world, reducing "minded" processes to algorithmic ones is definitely not thereby reducing them to material processes. All that this says is that wherever something mystical or badly understood seems to be involved in thinking (such as free will, or conscious deliberation for instance) one *should* be able to find an explanation where these unallowed terms are eliminated and at least a prospect is offered for understanding what happened in a deterministic way. There is of course no set recipe for this. One could go about actually describing a scheme in which there are several layers of nested loops of procedures (such as Johnson Laird's scheme for explaining consciousness in: Marcel and Bisiach (1988)) or one could speculate only about the kind of role these unexplained phenomena could be playing in the overall workings of the mind, and try seeing from this direction what kind of mechanisms could be involved (such as the General Workspace Hypothesis in Baars (1988)). However, and this is the important point, it is precisely the discovery of actual causal mechanisms underlying the processes that is not explicitly required in order to remain within bounds of accepted scientific practices. Turing-machines are nice partly for the reason - as I said in chapter one - that there are no constraints for providing causal mechanisms for an explanation to qualify for its being an explanation. It's just the *way* it works that's important not *how* it works. Finally, what then affords room for speculation about computational theories is that, on the other hand, due to Turing's invention, it seems that whatever is algorithmic is "computational".

When we turn to material processes, algorithms as tools play what I think are very different roles. Although I don't feel myself sufficiently prepared (nor entirely convinced) to argue this in considerable detail, especially because it touches on such big issues as the relationship between quantitative and qualitative aspects of the physical world; there are a number of reasons which I think may be contributing to the mess, about which I

shall - reluctantly - say a few words.

To put it briefly, it seems that our *everyday notion* of causality[40] is basically something like a flowchart. We think of the world in terms of events which cause each other or in terms of networks of things bearing certain "causal" relationships to each other. A description of a physical process, such as water coming to the boil will have in it reference to states of various entities in time, and the more detailed the description, the more elaborate the network. The desription of change in general will indeed look something like an FSA desription familiar from chapter two. The everyday notion of causality is suggestive of Chalmers' scheme as a deterministic sequence of state-shifts where the only difference between a flowchart description (or a Turing-machine computation) and the real thing is that in the real world the state-transitions are actually implemented. Causality is actual computation whereas Turing- machine computation is computation in the abstract.

And there is the conviction that matter is dumb. Although teleology has been eliminated by evolutionary theory the fact that matter changes in an organized fashion without preset goals (at least at the macro-level) is suggestive of "algorithms", because in the case of the mental it is algorithms which substitute for the eliminated homonculi and according to the above, causality is suggestive of algorithmicity anyway. It just seems like a small step to take, to actually feature algorithms in explanations (e.g. genetic algorithms).

To all this is added the fact that at this point only those physical processes are amenable to scientific investigation which are "computable" or at least have a mathematical model. There has been of late a surge of publications by physicists intended to convince the layman that the puzzles about mind and matter are in some sense equivalent (Penrose (1989) being one of them). These books usually contain extensive

---

[40] And it is a plausible view that there is no other notion. Bertrand Russell has argued in a classic paper that there is actually no scientific use for the concepts of cause and effect, what is more that there is no actual use of them either. These notions just do not appear in scientific theories, and that is because they are impossibly vague and any concrete employment means substituting something exact for the terms which will no longer have anything to do with the original concepts. Russell's arguments - to me at least - are entirely convincing. See: Russell (1959).

discussion of the differences between the determinism of the classical model of the physical world and the indeterminacy which is built into modern models. But the fact still remains that all models of physical processes will be algorithmic models, precisely because they are *models* of processes. Studying or predicting the outcome of physical processes often involve computer simulations, where sometimes properties of the simulation offer short-cuts to actually calculating various parameters, and hence offer an illusion of actual implementation. However, these are only models - what is actually there is a different matter.

All this is very vague, but it is intended as no more but just that, some vague hints on how algorithms come into the picture from the bottom side. The final moral to be drawn is that although one can get to algorithms from the bottom and one can get to algorithms from the top *the road travelled from the two sides will be a different one*, and these thoughts in and by themselves offer no justification for positing algorithms as the bridge between the two worlds. On the other hand, what I have said should suffice for providing at least the beginnings of an explanation of why the hypothesis *should seem* a natural one. It is in this sense that the transcendental (in the Kantian sense) goal that I have set for myself should have been accomplished. There should be more to this story of course: an explanation for instance (also a transcendental one) of how actual computational theories get constructed from the unseen assumptions. That story however is one that I hope has been told in part by preceding chapters. The separate transition to a "computational" framework in Dennett's sense, that is a computational theory which is supposed to explain neurology from propositional attitudes and *vice versa* - as I said in connection with the mental - is offered by the mathematical theory of algorithms. This essay was mostly about people who take up this offer and what happens to their theories as a consequence of this committment.

Conclusion

My aim in this essay has been to illustrate the insight that there is something wrong about what we think computations are, and also with the role it plays in various philosophical explanations concerning mentality. My other aim was to actually discover *why* there is something wrong with this. In this sense I aimed for something more conclusive than what Dennett says:

"These warring doctrines, High Church Computationalism [Dennett is referring to Fodor's theory. P.H.] and its many heresies, are not themselves theories; they are ideologies. They are ideologies about what the true theory of mind will or must be like, when we eventually divine it. Various attempts to create genuine theories - various research programs - *seem* to be committed to various ideologies arrayed in our space, but [...] the bond between research program and ideology is rather loose. In particular, the fact that great progress is (or is not) being made on a research program might tell us next to nothing about the ultimate soundness of its inspiring ideology." (In: Brand and Harnish (1986) p. 63.)

Although I agree with Dennett that CTM theories are essentially ideologies, there still remains the interesting question, that if they are *quite obviously* ideologies, what it is about the problem that makes the warring parties fail to realize this? How is it possible that the arguments advanced on either side claim that what they say provides conclusive evidence for one or another position? Ideologies sometimes think of themselves as ideologies, but these particular ones don't.

My answer was, that basically there is an illusion about how exact our notion of a computation is. On the one hand we have an elaborate and exact mathematical theory of algorithms, and this feeling of exactness tends to creep in to whatever notion of computation the ideologists tends to hold dear to their soul. I have tried to show that in

fact "being computational" is a vague notion[41] and one cannot transfer the rigidness of the formalism onto real life cases. On the other hand I also tried to offer some reason which could be behind the phenomenon that algorithms tend to appear cloaked in necessity in a place where their presence is actually unwarranted in a global way. Algorithms are tools in both physics and psychology, but the way they are tools does not warrant their assuming a position as a connecting agency.

I also wanted to show that Leibniz's challenge has in no way been answered. We haven't been able to come up with an argument to show that an artificially created mind is indeed possible, nor have we been able to discover what exactly is the difference between natural mechanisms and artificial ones - at least none of the arguments directed against CTM along these lines (or for it) seem to have been successful. The moral I would draw from all this is that right now we do not seem to have the conceptual tools to connect what we know about psychology (i.e. about ourselves) and what we know about the material world in any conclusive way. However, it also seems to me that if there is one thing which seems unpredictable, that is the *kind* of scientific discoveries that are going to be made in the future. Hence there is cause neither for panic-stricken attempts to explain away properties of the world as in David Chalmers work, nor is there cause for reading more of Heidegger (in this context) than is absolutely necessary. In other words there is no need to actually posit alternatives, if the problem is precisely that we do not yet *know* all the possible alternatives. What is important is trying to find those alternatives, especially if we have no way to be sure, on a priori grounds, that they do not exist.

---

[41] And I haven't even mentioned Church's thesis which implies that the notion is vague even in the mathematical sense. This does not mean that Turing-machines are not well defined.

BIBLIOGRAPHY

Armstrong, D.M. (1968). A Materialist Theory of the Mind. Routledge and Kegan Paul

Baars, B.J. (1988). A Cognitive Theory of Consciousness. Cambridge University Press

Brand, M. and Harnish, R. (1986). The Representation of Knowledge and Belief. The University of Arizona Press, Tucson, Arizona

Chalmers, David (1994). A Computational Foundation for Cognition. (ftp from Washington University)

Chomsky, N. (1972). Language and Mind. Harcourt Brace Jovanovich, Inc.

Chomsky, N. (1980). Rules and Representations. Columbia University Press, New York

Churchland, P. and Sejnowski, T. (1992) The Computational Brain. The MIT Press, Cambridge, Massachusetts

Clark, A. (1989). Microcognition. The MIT Press, Cambridge, Massachusetts

Cummins, R, and Pollock, J. (1991). Philosophy and AI. The MIT Press, Cambridge, Massachusetts

Damasio, Antonio. (1994). Descartes' Error: Emotion, Reason and the Human Brain. A Grosset/Putnam Book, New York

Dennett, D.C. (1987). The Intentional Stance. The MIT Press, Cambridge, Massachusetts

Dennett, D.C. (1991). Consciousness Explained. Little, Brown and Company, Boston, Toronto, London

Dennett,D. and Hofstadter, R. (1981) The Mind's I. Bantam Books, New York

Dretske, F. (1981). Knowledge and the Flow of Information. The MIT Press, Cambridge, Massachusetts

Dretske, F. (1985). Machines and the Mental. APA Proceedings

Dreyfus, H.L. (1972). What Computers Can't Do: The limits of artificial intelligence. Harper Colophon Books, New York

Fodor, J.A. (1975). The Language of Thought. Harvard University Press, Cambridge,

Massachusetts.

Fodor, J. A. (1982). Representations. The MIT Press Cambridge, Massachusets

Fodor, J.A. (1983). The Modularity of Mind: An essay on faculty psychology. The MIT Press, Cambridge, Massachusetts

Fodor, J.A. (1987). Psychosemantics: The problem of meaning in the philosophy of mind. The MIT Press, Cambridge, Massachusetts

Gordon, R.M. (1987). The Structure of Emotions: Investigations in cognitive philosophy. Cambridge University Press

Green, O.H. (1992). The Emotions: A philosophical theory. Kluwer Academic Publishers

Haugeland, J. (1985). Artificial Intelligence: The very idea. The MIT Press, Cambridge, Massachusetts

Hofstadter, D.R. (1979). Gödel, Escher, Bach: An Eternal Golden Braid. Harvester Press, Sussex

Hookway, C. ed. (1984). Minds, Machines and Evolution: Philosophical Studies. Cambridge University Press, Cambridge, Guess Where...

Hopcroft, J. and Ullman, D. (1969). Formal Languages and their Relation to Automata. Addison-Wesley Publishing Company

Johnson-Laird, P.H. (1988). The Computer and the Mind: An Introduction to Cognitive Science. Harvard University Press, Cambridge, Massachusetts

Leibniz, G.W.F. (1951). Selections. Charles Scribner and Sons, New York

Lewis, D. (1972). Psychophysical and Theoretical Identifications. Australasian Journal of Philosophy 50: 249-58

Looneburg (von), W(ilfred) F(rancis). (1992). My Mind is a Machine: Somber Reflections on Thinking. Amsterdam, The Hague.

Marcel, A.J. and Bisiach, E. eds. (1988) Consciousness in Contemporary Science. Clarendon Press, Oxford

Martin, C.B. (1993). What is Imagistic about Verbal Imagery and Why Does It Matter? Forthcoming.

McNamara, P. ed. (1993). Which Computers Can Think (If Any)? Philosophical Studies Vol. 73 No. 2. Kluwer Academic Publishers

Pagels, H. R. ed. (1984). Computer Culture: The scientific, intellectual and social impact of the computer. The New York Academy of Sciences, New York, New York

Putnam, H. (1975). Mind, Language and Reality. Cambridge University Press

Putnam, H. (1988). Representation and Reality. The MIT Press, Cambridge Massachusetts

Penrose, R. (1989). The Emperor's New Mind: Concerning computers, minds and the laws of physics. Vintage, Oxford University Press, New York

Pylyshyn, Z. W. (1984). Computation and Cognition: Toward a Foundation for Cognitive Science. The MIT Press, Cambridge, Massachusetts

Russell, B. (1959). Mysticism and Logic. George Allan and Unwin Ltd. London, England

Searle, J.R. (1983). Intentionality: An essay in the philosophy of mind. Cambridge University Press

Searle, J.R. (1992). The Rediscovery of the Mind. The MIT Press, Cambridge, Massachusetts

Slezak, P. and Albury, W.R. eds.: Computers, Brains and Minds: Essays in cognitive science. Kluwer Academic Publishers, 1989

Thalberg, I. (1977). Perception, Emotion and Action. Oxford: Basil Blackwell

von Neumann, J. (1958). The Computer and the Brain. Yale University Press

Zajonc, R.B. (1980). Feeling and Thinking: Preferences need no inferences. American Psychologist Vol. 35, No.2, 151-175