Roel Wieringa

Machine Intelligence and Explication

Rapport nr. IR-129

VRIJE UNIVERSITEIT AMSTERDAM

Faculteit Wiskunde en Informatica, De Boelelaan 1081, 1081 HV Amsterdam
Machine Intelligence and Explication

Roel Wieringa

Department of Computer Science
Free University
Amsterdam
## CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Machines and explicitness</strong></td>
<td></td>
</tr>
<tr>
<td>1.1 Systems, states, and processes</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Machines, Turing machines, computers</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Explicit descriptions</td>
<td>5</td>
</tr>
<tr>
<td><strong>2. The mechanization of explication</strong></td>
<td></td>
</tr>
<tr>
<td>2.1 The Turing test</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Assumptions of the Turing test</td>
<td>9</td>
</tr>
<tr>
<td>2.3 Human intelligence and the ability to explicate</td>
<td>11</td>
</tr>
<tr>
<td>2.4 Explication by machine</td>
<td>13</td>
</tr>
<tr>
<td>2.5 Artificial intelligence and machine intelligence</td>
<td>17</td>
</tr>
<tr>
<td><strong>3. Comparison with other viewpoints</strong></td>
<td></td>
</tr>
<tr>
<td>3.1 Turing</td>
<td>18</td>
</tr>
<tr>
<td>3.2 Jefferson</td>
<td>21</td>
</tr>
<tr>
<td>3.3 Semantic information processing</td>
<td>21</td>
</tr>
<tr>
<td>3.4 Searle</td>
<td>30</td>
</tr>
<tr>
<td>3.5 Digression: Implementation-independence and AI methodology</td>
<td>34</td>
</tr>
<tr>
<td>3.6 Dreyfus</td>
<td>35</td>
</tr>
<tr>
<td><strong>4. Some consequences of the thesis</strong></td>
<td></td>
</tr>
<tr>
<td>4.1 Explication revisited</td>
<td>38</td>
</tr>
<tr>
<td>4.2 Ontological monism and epistemological dualism</td>
<td>41</td>
</tr>
<tr>
<td>4.3 Dualism between man and machine</td>
<td>44</td>
</tr>
<tr>
<td><strong>5. Summary and conclusion</strong></td>
<td></td>
</tr>
<tr>
<td>5.1 Summary</td>
<td>47</td>
</tr>
<tr>
<td>5.2 Conclusion</td>
<td>49</td>
</tr>
<tr>
<td><strong>References</strong></td>
<td>50</td>
</tr>
</tbody>
</table>
PREFACE

This report is an MA ("doctoraal") thesis submitted to the department of philosophy, university of Amsterdam. It attempts to answer the question whether machines can think by conceptual analysis. Ideally, a conceptual analysis should give plausible explications of the concepts of "machine" and "intelligence" and then investigate the intersection of the sets of entities defined by these explications. If the intersection is empty and the a priori argument is correct (or plausible), then empirical research into machine intelligence will (plausibly) not result in an intelligent machine. On the other hand, if conceptual analysis cannot show the intersection to be empty, it remains an empirical (or rather, technical) question whether such machines can actually be constructed.

Such a neat argument cannot be produced, however, due to the vagueness of the concept of intelligence. It is quite possible to provide a rather uncontroversial explication of the concept of machine. Existing controversy about the possibility of machine intelligence is about the nature of intelligence, not about the nature of machines. Indeed, if intelligence could be unambiguously defined, we could (in principle) build a machine to implement it. Those who believe that intelligence cannot be realized in a machine, cannot base their arguments on an explicit and uncontroversial analysis of the concept of intelligence.

The argument in this essay therefore follows a different route than the ideal argument. After a definition of machine which combines the important characteristics of that concept in computer science and systems theory, I try to explicate why we think this definition captures our informal intuitions about the nature of machine-like, mechanical processes adequately. This leads to an explication of what explicitly described processes are.

Chapter 2 then replaces the question whether machines can think by the simpler question whether machines can explicate. Using the explication of the concept of explicit descriptions given in chapter 1, I argue that the process of explication cannot be explicitly described. If that argument is correct (or plausible), then no machine can (plausibly) be built which explicates a situation, for to build a machine is to implement an explicit description. The bearing on the original question of machine intelligence is this: If human intelligence presupposes the ability to explicate, an entity which cannot explicate cannot have human intelligence. Chapter 2 contains some arguments why we do not attribute human intelligence to a being which cannot explicate.

In chapter 3 the argument is defended against some possible counterarguments and compared with two well-known criticisms of artificial intelligence, those by Dreyfus and Searle. Finally, chapter 4 explores some practical as well as metaphysical consequences of the thesis.

This short overview of the argument should already have made clear that I do not believe that an uncontroversial, explicit proof of the impossibility of machine intelligence exists. If such a proof existed, it could be automated, which would be close to a refutation of what the proof would establish. It follows that holes can be shot in the argument. It won't execute without errors in all environments. Therefore, in the interest of (among other things) brevity, I stopped explicating when further explication would backfire and merely expose the emptiness of the argument. That the empty argument would have been, as Isshuu Miura said, closer to the truth than the essay I wrote now. But then, I wouldn't have passed the exam by handing in an empty paper.

Working on this thesis made me painfully aware that the semantic network we live in is essentially fluid and unbounded. Thanks are due to my supervisor Hans Swart, who followed me on my wanderings in various interesting directions and who suggested I stay with one topic and work that out. Thanks are also due to Dick de Jongh and Loet Leydesdorff, who gave constructive criticisms of the thesis.

Amsterdam, june 25, 1987
R.J.W.
The movements start from the abdominal parts and the breath passing through the teeth produces various sounds. When articulated they linguistically make sense. Thus we clearly realize that they are unsubstantial. Rinzai (Lin-Chi d. 867) in: D.T. Suzuki [1960], p. 42.

Of course, what I have been saying all this while is just a part of the confusion of sounds of which the world is so full. Isshuu Miura in: Miura & Sasaki [1965], p. 45.

Actually, the task of capturing the meaning of data is a never-ending one. T. Codd [1979], p. 398.
Chapter 1
Machines and explicitness

The concept of a machine has been explicated in automata theory (Minsky [1967]), the theory of formal languages (Hopcroft & Ullman [1969]), and the foundations of mathematics (Davis [1958]). The explication given in this chapter follows Minsky and starts from the concept of a system.

1.1. Systems, states, and processes

A system is any part of reality we are interested in and of which we can indicate the boundaries. The system boundaries distinguish the system from its environment, which interests us only in so far as it affects the system or is affected by the system. The influence of the environment on the system is called the system input, and the influence of the system on its environment is called the system output. I use the term "external influence" as synonymous with "system input" and "system behavior" as synonymous with "system output."

When we direct our attention toward a system of which we cannot observe the internal structure, we are limited to observing the history of inputs to and outputs of the system. Suppose \( H_1(t) \) and \( H_2(t) \) are two possible histories of the system at time \( t \). Then the system can distinguish between them iff there is a possible sequence of inputs after \( t \) which would lead to one output sequence for \( H_1 \) and a different output sequence for \( H_2 \). The system cannot distinguish between \( H_1(t) \) and \( H_2(t) \) iff for every possible future input sequence the same output sequence would ensue when that sequence follows \( H_1(t) \) as when it follows \( H_2(t) \).

It is easy to see that indistinguishability of histories is an equivalence relation. It is reflexive, symmetric and transitive. We can therefore partition the set of all possible histories at time \( t \) in an exhaustive and disjoint set of equivalence classes. The equivalence class of the history at time \( t \) is called the state of the system at time \( t \) (cf. Minsky [1965], pp. 11 ff.). Giving the state at time \( t \) is to give structural information about the system, for it captures exactly what it is about the internal configuration of the system that makes a difference for future behavior. The state of a system is a memory of previous events, since it contains information about the history of the system. It contains exactly the information which is relevant to every possible future of the system.

A process is a sequence of states. When a system goes through a process, it carries out or executes the process. I will use the two terms "go through" and "carry out" as synonymous in this context. Note that a process is a sequence of states, not of inputs and outputs. A system may receive no input and produce no output for some time and still carry out a process, i.e. change state. The concept of state puts us beyond behaviorism.

Up till now, the system has been treated as a black box. If we were to open it, we could use the pattern of state transitions of the system to identify subsystems. The state of the system may then be decomposed into the state of its subsystems plus a summation rule such that each subsystem goes through a process of its own, with respect to which the rest of the system acts as an environment. For example, a copper wire may be considered as consisting of a series of resistors such that its resistance, which is a quantity characterising part of its state, is the sum of the resistances of the parts. Or a gas may be considered to consist of particles such that the quantities characterising its state are related to those states of the particles in a statistical way.

It is important to see that in each of these cases the subsystems can be treated in exactly the same way as the whole system: Without looking at or even knowing their internal structure, they can be described as going through a sequence of states, displaying behavior, subject to environmental conditions. Much of science progresses in this top-down manner.

---

\[1. \text{If and only if.}\]
If a system can be quantitatively described, its states can be characterised by a set of numbers which are values of its state variables. State variables change during a process and stand in certain relations to each other. If the system is divided into subsystems, its state variables also stand in relations to the state variables of its subsystems. The name of a state variable intuitively indicates the role it plays in the state changes of the system. The definition of a state variable is functional, i.e. it indicates the function which the variable has in the determination of the next state.

The number of state variables of a system is called its dimension. The state variables define a state space in which coordinate axes can be defined corresponding to each state variable. Any process which the system goes through can be visualised as a path through the state space. Some systems are infinite-dimensional. For example, when a piece of metal is magnetised, the next degree of magnetisation is dependent not only on the current degree of magnetisation and current input, but on the whole history of magnetisation and the current input. Because magnetisation is a continuous process, infinitely many numbers are needed to determine the next state of the system. The state space is therefore infinite-dimensional.

For every system we can try to find the state transition function, which determines the next state and output from the current state and input. Systems can be classified according to the type of state transition function they have. The most important distinction is that between continuous systems, which have a continuous next state function, and discrete systems, which have a discrete transition function. This difference concerns the kind of mathematics used to do next state computations, continuous or discrete. I will talk of state evolution when I want to leave open whether I am talking about continuous or discrete systems, and about state transitions when I am talking about discrete systems only.

Another distinction distinguishes deterministic systems, which have a deterministic state transition function, i.e. a function which assigns one next state to every current input & state, from nondeterministic systems, which have a nondeterministic next-state function, i.e a function which assigns a set of next states. A special kind of nondeterministic systems are probabilistic systems, whose next-state function is non-deterministic and gives the probability of each state in the next-state set.

In the sequel, the distinctions between discrete, continuous, deterministic, nondeterministic and probabilistic systems play no role. I mentioned them only to draw attention to the fact that if my argument is valid, it is valid for all these systems. It is irrelevant whether we are talking about computers, connection machines, Boltzmann machines, neural networks, massively parallel architectures, or any other type of machine, as long as it is a machine.

1.2. Machines, Turing machines, computers.
A machine is a system for which there is a state evolution function. Put this way, the concept of a machine does not seem very helpful, since according to it there may be systems of which there is a state evolution function which is unknown to us. There is a state evolution function of such systems, but we don’t know it.

However, this definition of ”machine” agrees with the usage in modern science. Scientific research is aimed at discovering which systems found in nature are machines and it shows that they are machines by producing state evolution functions of them which fit observable

---

The assumption that a system is a machine is a justification for the rationality of scientific research, but must itself be justified by the results of scientific research. This is a kind of hermeneutical circle which is broken by investigating systems which ”look promising.” The assumption that nature is a machine is the distinguishing characteristic of modern, i.e. post-Newtonian, science. Dijksterhuis concludes his [1961] as follows:

“In those days [the 17th century] an entirely new standpoint towards nature has been reached: ’substantial’ thinking, which inquired about the nature of things, had to be exchanged for ’functional’ thinking, which wanted to ascertain the behaviour of things in their interdependence; the treatment of natural phenomena in words had to be abandoned in favour of a mathematical formulation of the relations between them.”

(p. 501)
behavior. It produces them in the sense that Johnny produces a dinky toy from his pocket. If a system is a machine after scientists find a fitting state evolution function for it, it was a machine before they found it. Questions about the ontological status of universals aside, we can capture the gist of the concept of a machine by saying that a machine is a system for which there is a state evolution function.

The question whether a system is a machine thus reduces to the question whether there is a state evolution function of that system, and we can give a positive answer to that question by producing such a function. Put differently, to be a machine is to be describable as a machine.

This definition does not exclude things which any definition of "machine" should include, e.g. radios, cars, computers and airplanes. It also includes natural systems we know how to describe as a machine, such as the solar system, a falling stone, subnuclear particles (to a certain extent) and even man.

To substantiate this last example, consider a library information system in which the user's actions are: To borrow a book, to return a book, to lose a book, to pay for a book, to extend a loan, and to request a book. A simple finite-state transition diagram can be drawn in which these actions figure as state transitions. Under this description, the user is a machine. Needless to say, this description does not describe his intelligence. Moreover, similar descriptions can be given of the books in the library, which thereby are machines as well.

The definition of machines thus includes things which we would not regard as machines at first sight, even though we describe them as machines. On the other hand, it excludes systems of which we can reasonably say that they are not machines. The weather, two people talking to each other, a writer writing a novel, the processes reported to go on during the !Kung Bushmen healing dance (Katz [1982]), hopping during transcendental meditation, and subnuclear "particles" whose behavior is dimly understood are not machines. And if we find a fitting description of the evolution any of these systems, our intuitions about that system would probably change in such a way that we regard it henceforth as a machine. This is extra evidence that the definition in the first section of this section adequately explicates our intuitions.

Scientists attempt to describe some of these dimly understood systems as machines while others are written off, without argument (at least not with an argument with the size of this essay), as not explicitly describable by a state evolution function. Often, even the reports about the existence of such processes are doubted. But we may try to show of any dimly understood process that it is not a machine by showing on a priori grounds that there cannot exist a state evolution function of it. The conceptual analysis must then use the essential characteristic of the concept of a state to show that that concept is not applicable to the process in question. This thesis tries to give such an argument for one type of process, explication.

The examples and counter-examples given so far concern only concrete machines. Examples of abstract machines are the finite-state automata, push-down store machines, linear-bounded automata and Turing machines of abstract automata theory. Every computer program is an abstract machine, which is simulated by a concrete computer. These examples emphasize that a machine is a conceptualization. Concrete machines such as a radio or the solar system exist independently of us, but we cannot call them machines when we do not have a description of their state evolution function.

An interesting consequence of this account is that a broken-down radio is not a machine. It is a system of which we don't know the state evolution function. It behaves erratically. In that case, we use the state evolution function of a functioning radio as a reference point against which to measure the performance of the broken radio. Repairing the broken radio is making it function according to that state evolution function. It is the state evolution function which is the heart of the machine, not the hunk of metal and plastic emitting sounds on command.

This procedure contrasts with that of natural science, where in case of a persistent and important conflict between description and behavior, we try to fix the description, not the 3. "His" abbreviates "his or her."
system. The library user borrowing a book and then refusing to execute any of the actions of the relevant finite-state transition diagram, acts like a broken radio and is acted upon in the same way: External force is used to make him behave properly.

One class of machines, Turing machines, plays a central role in discussions about MI. Some terminology pertaining to this class is briefly reviewed below. Very little of this will play a role in the rest of this essay, and I mention it, as before, to point out that my argument applies to all these different types of machines.

A Turing machine (TM) is a discrete system with two components, one of which can be in a finite number of states, and the other of which is a tape consisting of an infinite number of squares, each of which can store one symbol from a finite alphabet. The tape plays the role both of the system environment and of extra memory (in addition to the memory provided by the states). The finite-state machine and tape communicate via a read/write head. The TM is started with the head scanning one square. The state transition function defines, for every current state and scanned symbol, the next state, the symbol to write on the current square, and whether to move one square to the right or left.

The Turing-Church thesis (TC) says that a TM can be used to model any computation which can be carried out by symbol manipulation. One way to see this is that every symbol manipulation process carried out by man consists of a series of replacements of (sequences of) symbols by (sequences of) symbols written on a medium, e.g. paper. A TM does just that, not hampered by lack of paper, ink, time, or energy and not plagued by fatigue, illness, untimely death, various distractions, infatuation, anger, or boredom.

The TC thesis is a thesis which is well-argued but not proved. Whenever intuitive concepts are formalized by exact concepts, the question whether the formal concept exactly covers the intuitive concept cannot be settled by proof. Proofs start with definitions, and cannot justify that the definitions proofs start with explicate our informal intuitions "correctly." One can give arguments by conceptual analysis and by pointing out successes of the definition (e.g. the consequences which follow from the definition are intuitively satisfying), but there is always room for disagreement.

A universal TM (UTM) is a TM such that, given a description of the transition function of any TM in a certain format on its tape, and the initial configuration of the tape of that TM, it simulates the process the TM would go through (to simulate one state transition of the TM, it goes through several transitions itself). The description of the TM on the tape of a UTM is called a TM program.

In the world of computers, different, but for our purposes synonymous, terminology is used. A TM is there called a special purpose digital computer and a UTM is called a general purpose digital computer, or simply computer. A TM program is called a computer program or simply program. For most purposes we can use the terms "program" and "TM" interchangeably.

By treating a computer as an (abstract) UTM, we disregard space- and time bounds of the concrete machine. The tape of the UTM is the computer's memory, the finite-state part of the UTM is the computer's central processing unit (CPU). This abstraction from the concrete machine proved to provide better understanding of what a computer does than modelling by another type of abstract machine.

The machines defined thus far are deterministic. Non-deterministic and probabilistic versions exist as well.

Finally, a special purpose continuous computer is a deterministic continuous system. These are usually called analog computers. To the concept of a general purpose digital computer corresponds a general purpose continuous computer, a machine which simulates the (continuous) processes which any continuous deterministic system can go through. In practice, continuous general purpose computers are never built because reliable continuous memories, which should be able to store and distinguish among an infinite set of symbols (eg. a set of real numbers) are extremely difficult to construct (see eg. Hollingdale & Tootill [1965], p.72). Moreover, special purpose continuous computers can be simulated by general purpose digital
computers to arbitrary high precision, so we do not need a general purpose continuous computer to play the role of universal machine in the continuous realm.

1.3. Explicit descriptions

In the previous section I defined machines as systems whose behavior is described by a state evolution or transition function. In this section I want to investigate three properties of these descriptions which will play a role in the argument of chapter 2. It will be argued there that explication is a process which cannot be described by a description having any of these three processes.

The three properties are repeatability, context-independence, and communicability. I will call descriptions which have those properties explicit descriptions.

A process is repeatable if it develops in the same way whenever the same state occurs, followed by the same sequence of inputs. For example, any TM will go through the same state changes whenever initialized with the same input on its tape and started in the same state. Similarly, if at times $t_0$ and $t_1$ the solar system passes through the same state and following $t_0$ and $t_1$ the same sequence of inputs follows, it will go through the same state evolution. This simply follows from the definition of the concept of a state. The use of the word "same" is crucial here. TM's go through the same state changes by definition and the solar system by causes outside our control, but in both cases there is a notion of "the same state" which warrants use of the notion "the same state sequence" = "the same process." The notion of "the same state" consists of certain state descriptions being applicable at different places and moments. 4

The use of the word "same" thus emphasises the conceptual nature of machines. Machines exist independently of us, but they are only machines under a description. For example, at a suitable level of abstraction, the solar system is a machine, but at a higher level of detail the processes become chaotic, i.e. they are subject to random fluctuations or their pattern is simply unknown. As another example, a library user is a system executing different processes concurrently, at least one of which, his behavior as a library user, can be explicitly described. At suitable levels of abstraction, some of the other processes can be described as machines as well, but when we get down to lower levels of detail, the processes which we describe lose their characteristic of repeatability. The repeatability of the process corresponds to the repeatable applicability of the process description and I will often loosely speak of "repeatable process description," where I should speak of "repeatably applicable process description."

The second characteristic of explicit descriptions is context-independence. If a process is repeatable, its description is context-free in the sense that the influence of the environment on its state evolution is specified exhaustively. A system may execute more than one process concurrently (e.g., my body is executing many processes at once) or it may be described in many different ways as executing one process at different levels of abstraction, but given a description at a certain level of abstraction in the form of a state evolution function, the effects of the environment upon the described process is exhaustively specified. By implication, what is not specified in it, has no influence upon the evolution of the process at that level of abstraction. Placed in a context which satisfies the properties required by the state evolution function but different in any other property, the system will execute the same process.

The third characteristic of explicit descriptions is that they are communicable. A repeatable process must be identifiable as the same process every time it occurs. In whatever way this identification takes place, a group of people must agree that the process is identified and must know the process by its identification. In other words, the process description is communicable to members of a group who can apply it, i.e. use it to identify a process. This group is not limited by vagaries of birth, status, culture or race, but by its competence to understand and apply the process description. Verification that a member of this group possesses this competence is done

---

4. This account can be adapted for non-deterministic and probabilistic cases by replacing "state" by "set of states" or "set of states with a probability distribution," respectively.
by other members of the group, who must be competent themselves as well. The most common institution where this happens is in our schools and universities. Apart from these tests of competence, the group of competent people throws up no barriers to aspiring members. There may be other barriers, such as an aspiring member not being able to learn how to apply the process descriptions, or the admission fee to the institution of learning being exhorbitantly high, or the government instituting a bizarre lottery system to decide who is going to be allowed to join a school or university. But in all cases, the requirement of applicability of the description by different people ensures that the description does not refer to properties of the person who applies it.

In short, an explicit description is applicable in different contexts by different people to a system which does not depend upon the idiosyncracies of the human being who happens to apply the description. No intuition is required to apply it, since all relevant aspects of the contexts are mentioned in it. In the case of abstract machines like TM's, applicability by any person is ensured by making the parts of the description as simple as possible (but the total description may be very complicated).

Every machine is an explicitly described system. The converse may be true as well, i.e. every system executing an explicitly described process is a machine, but that is not of importance to my thesis. My argument is that a machine cannot explicate because explication cannot be explicitly described. If a machine is an explicitly described system, then if explication cannot be explicitly described, explication cannot be carried out by machine.
Chapter 2
The mechanization of explication

The second half of the answer to the question whether machines can think should ideally consist of an explication of the concept of intelligence. Due to the vagueness of that concept, this is not possible in an uncontroversial way. One way to circumvent this problem is to replace the question whether machines can think by the question how we could find out whether machines are intelligent. Turing [1950] did this in his famous test. The Turing test (TT) has been the stage for discussions about the possibility of MI ever since and is discussed in section 2.1. Section 2.2 criticizes the validity of the test as a correct way of settling the answer to the question whether machines can think. This criticism is also a motivation for the structure of my own argument, which is then presented in sections 2.3 and 2.4. Section 2.5 clears a possible misunderstanding about what is actually argued in the foregoing sections.

2.1. The Turing test
There are four versions of the TT.

In the first version (TT₁) an interrogator, C, is connected by a data communication line to two people A and B with a terminal (a keyboard and a video screen). He has to find out, by asking questions to either one and by reading the answers, which of the other two is the man and which is the woman. A, B, and C can each see what the others type in. Now, without knowledge of the interviewer, A is replaced by a machine designed to answer questions in natural language. Turing proposes to replace the question "Can machines think?" by the question "Will the interrogator decide wrongly who is a man and who is a woman as often in the case that A is a machine as in the case that A and B are both human beings?"

Version two of the test, TT₂, is the one usually discussed by philosophers. In TT₂ C knows that either A or B is a machine and is to decide who is the machine and who is the human being.

In version three, TT₃, specialists compare, in a peer review, the performance of a group of specialists in their area. Without their knowing it, among the group to be judged there is a machine built to perform on a specialist level. The test is to see if the machine is judged significantly different from human specialists. The MYCIN expert system was subjected to this test (Yu et al. [1984]).

In TT₄ the performance of the specialist machine is compared with that of human specialists to see if it significantly differs from them. MYCIN was subjected to this test as well (Yu et al. [1979]), before it was recognized that TT₃ is more fair to the machine.¹

The last two tests are versions of the first two in the domain of expert systems and are not relevant to the discussion in this essay. The first two are relevant, so we have to see if there is any difference between the two and if so, which is the "correct" one.

TT₁ is designed to test whether we can make machines perform in such a way that we do not need to know whether we are interacting with a person or a machine. The hypothesis whether A is a machine does not enter C's mind, unless during the interview an explicit reason arises which forces him to do so. Under the circumstances of TT₁, the hypothesis does not cross C's mind, because it is too absurd in everyday conversation to take the idea seriously. If someone does find it necessary to proceed with a conversation upon the explicit assumption that the other is a machine, we gently suggest mental treatment (cf. Robinson [1972]). Because C does not know that he has to decide whether he is talking to a machine or a human being, he does not have to explicate any of his assumptions about the behavior of human beings either. This

¹. On the other hand, TT₄ is fairer to the prospective buyer of the machine, who wants it to clearly outperform its human counterparts.
has two advantages. First, the instructions to C do not have to mention any supposed difference between man and machine. This is consistent with either outcome of the test. Second, the test can be described and judged without having to mention distinguishing characteristics of human intelligence (which the machine playing the role of A is or is not going to display). Instead, C uses his implicit, unexplicated knowledge of what it is to behave with human intelligence. That way, the test can be discussed before we can actually build a machine to play the role of A.

In $TT_2$, the situation is very different. C will now use some of his (right or wrong) ideas about what the crucial difference(s) between man and machine are. This is harder on the machine, since we now test, not whether we can fool C, but whether C can detect any difference between the behavior of the human being and machine he is interviewing. For example, when A starts uttering gibberish, C will decide that A is a machine and that its language generating routine has broken down. In the case of $TT_1$, C would have assumed that A is a foreigner switching to his own language, or is banging his fist on the keyboard, or that the communication line has broken down or anything but the hypothesis that A is a machine. Since B and C can be any person, $TT_2$ tests whether the machine A behaves indistinguishable from any person for any observer. For an interviewer who knows that one of the rooms contains a machine and the other a human being, the (linguistic) behavior of a machine passing $TT_2$ is indistinguishable from that of a human being, while a machine which passes $TT_1$ could still display behavior which in $TT_2$ would be distinguishable from that of human beings. Whereas $TT_1$ is a test of C (viz. whether we can fool him), $TT_2$ is a test of A (viz. whether it behaves indistinguishably from a human being). From this comparison I conclude that $TT_2$ is the right explication of the test. When in the following I refer to TT, I mean $TT_2$.

We can explicate the TT semiformally as the following definition.

A is intelligent := $\forall B, C \in \text{persons} [C \text{ cannot distinguish } A \text{ and } B \text{'s behavior}]$ (*)

The definition is semiformal because the concepts of person and indistinguishable behavior are not defined formally (and, as this thesis implies, cannot defined formally).

AI enthusiasts sometimes forget the quantifier on C and claim intelligence for a system which could fool one interviewer (for some of the time). But if we view (*) as an operational definition of "intelligence," we must include the quantifier on C. Similarly, it is essential to the definition that we cannot in advance single out a class of human beings distinguishable from machines and restrict quantification on B to those we cannot distinguish from machines.

Two other essential elements of the definition are indistinguishability and behavior. Which behavior do we look at, and when are two instances of those behaviors indistinguishable? As regards the first question, remember that any library user is a machine under a certain, very abstract, description. At that level of abstraction, we don't need a TT to decide whether there is a machine which thinks, for every library user is one. Neither do we need the level of detail at which the behavior of individual neurons is described, for then C's role as a supplier of intuitive judgements of indistinguishability would be superfluous. The second question, when two behaviors are to count as indistinguishable, must be decided on intuitive grounds. Plainly, no two behaviors are exactly indistinguishable. In physics, where explicitly described behavior is

2. There is an interesting analogon between the TT extended to the level of neurons and the sorites paradox.

Compare:
One grain of sand is not a sandheap.
If $n$ grains of sand are not a sandheap, $n + 1$ grains of sand are not a sandheap either.

(By induction on $n$:) $\forall n \in \mathbb{N} [n \text{ grains of sand are not a sandheap}].$

If one neuron of a person P is replaced by a machine with indistinguishable behavior, P still thinks.
If P thinks after $n$ neurons have been replaced, then P still thinks after $n + 1$ neurons have been replaced.

P thinks after all his neurons have been replaced.
measured quantitatively, use is made of an elaborate theory of measurement errors and sampling in order to average out individual but insignificant differences between different behaviors. If we had measurements to which such a theory could be applied in the case of the behavior relevant to the TT, then the TT, and C's role in it as a supplier of intuitive judgements, would be superfluous. I conclude that the relevant behavior is the type of behavior with which a human being intuitively feels at home and that indistinguishability is defined with respect to this intuition.

The TT has now been explicated as a test in which any interviewer cannot intuitively distinguish the (linguistic) behavior of any person from that of a machine.

### 2.2. Assumptions of the Turing test

I think the TT is attractive to many people because it makes two misleading (and related) mistakes:

1. The question whether a machine can actually be constructed to pass the test is taken out of the conceptual realm of a priori arguments. In the hands of Turing, it becomes not even an empirical question but a technical one. This creates an open atmosphere in which researchers are not bridled by dogmas or ideologies fixed by church or state and have the freedom to say "Let's try it. Some day we may succeed." (cf. Turing [1950], p. 23). This appeals to our sense of freedom of mind. 3

2. When we want to give arguments why an entity is intelligent, we are ultimately led back to observable behavior. If we want to deny intelligence to a machine which behaves indistinguishably from a human being, then we should deny intelligence to the human being as well (Turing [1950], p. 17). This appeals to our sense of intellectual honesty: Once we have decided to be open-minded, we should be willing to support our statements by arguments and our arguments by facts, and in this we should not give favors to one party.

These two moves in the argument are mistaken because they distract the discussion about the question of MI from the real issue. First, the question whether machines can think is a conceptual one. The shift of attention from "what is intelligence" to "how do we know that X is intelligent?" is inappropriate. It has been repeatedly asserted by different authors (Robinson [1972], Searle [1980a & b]) that the issue is not how we know that people are intelligent, nor whether we can give arguments for the statement that someone has human intelligence, but what it is we attribute to a human being when we attribute normal human intelligence to him or her. We may be mistaken in the attribution, or we may give incorrect arguments in support of a correct attribution, but the elucidation of what it is we attribute can be carried out independently of the investigation of when we are mistaken and of examination of the kind of arguments we give in support of the attribution. From this it follows that the first move made by the TT can be countered by noting that the last two questions (when are we mistaken in the attribution of intelligence and what kind of (right or wrong) arguments do we give in support of it) are empirical but that the first, elucidation of what it is we attribute, is conceptual. The same observation also counters the second move. We are not asking how we know that other beings are intelligent, we are asking what intelligence is. If we can give a plausible explication of the concept of intelligence as applied to human beings, we may find that it is the kind of property which cannot logically (or plausibly, depending upon the hardness of the argument) be attributed to a machine. Viewed in this way, the TT tests whether we are mistaken in the attribution of human intelligence.

This point is not generally appreciated in the AI community. For example, Minsky [1982] predicts that with the advent of smarter machines, the meaning of "intelligence" will evolve

---

3. Some believers in the possibility of MI say that the unbelievers ought to show more humility and admit of the possibility that there are intelligent machines, that these can -*gasp*- even be more intelligent than human beings. This appeals to a sense of freedom of mind and a room for bold ideas. An equally emotional argument from the other side is that one ought not to try to imitate God and create beings who may have a soul. These discussions regress into insinuations about each other’s lack of openness of mind or religiosity.
until it is appropriate to apply it to machines as well as to people. This ignores the possibility that the meaning of "intelligence" as it is currently used may contain a valid distinction between man and machine (cf. Robinson [1972]). If there is, it can only be uncovered by conceptual analysis. It can also get blurred by a shift in meaning of "intelligence," but in that case we would have to find a new term to indicate the distinction. What is at issue is whether the distinction is valid. Note that, curiously, appeal to a shift in meaning of the word "intelligence" is a reversion of the movement from the conceptual to the technical realm. It is merely a conceptual matter whether machines can be intelligent, Minsky says, and we should have the openness of mind to go through this change.

The confusion of the question what human intelligence is with the question how we know that someone has human intelligence did not start with Turing's original article. Jefferson, to whose Lister oration Turing responded, traces the opinion to Descartes:

If, [Descartes says] one had a machine that had the shape and appearance of a monkey or other animal without a reasoning soul (i.e. without a human mind) there would be no means of knowing which was the counterfeit. (Jefferson [1949], p. 1106)

Descartes said it is possible to produce such a counterfeit of an animal but not for man, since man has a mind. Having a mind is the essential difference in the Cartesian dualistic universe between man and the rest of the world. Turing wrote his article to deny the impossibility of constructing a counterfeit human being. If we have such a counterfeit human being, Turing says, we must view it simply as a human being because it is a human being. I argue against this that it is impossible to produce a counterfeit human being (and perhaps of other animals as well) without assuming a dualist metaphysics. The argument in section 2.4 against the possible existence of intelligent machines does not require the assumption that any intelligent being must have a soul.

Turing also considers the claim that the only way to know whether a machine thinks is to be that machine (Turing [1950], p. 17). Since this claim is applicable to human beings and machines alike, it leads to solipsism. This is a difficult position to maintain in everyday life, and "instead of arguing continually over this point it is usual to have the polite convention that everyone thinks." (Turing [1950], p. 17) It sure is usual to have that convention, but this does not show that we can extend that convention to machines. The two questions "What is human intelligence?" and "How do we know that an entity possesses human intelligence?" are related but distinct.

In short, the TT can show at most that we can be mistaken in the attribution of human intelligence. It does not show anything either way about the possibility of machine intelligence. On the other hand, from a conceptual analysis of MI we can expect one out of two things: Either the concepts of machine and intelligence are compatible, or they are incompatible. In the first case, technical research will have to answer the question whether we can actually build such a machine, and empirical research will have to answer the question whether such a machine exists in nature. For example, empirical research could try to show that man is such a machine. (In section 2.5 I will distinguish natural machines, studied by natural science, from artificial machines, studied by engineering science.) In the first case, technical and empirical research may or may not succeed. But in the second case, technical research cannot result in the construction of a thinking machine and research in natural science cannot result in the discovery of a thinking machine (e.g., man). This essay tries to make it plausible that the second is the case. The argument has the following structure.

(1) If A has human intelligence, A can explicate.
(2) If A is a machine, it cannot explicate.
(3) If A is a machine, it does not have human intelligence.

The next two sections provide arguments in favor of each of the premisses (1) and (2).
2.3. Human intelligence and the ability to explicate

An explicit description of a state of the world (at a certain level of abstraction) enables anyone of an arbitrary group of people to identify the state whenever it occurs, in whichever context it occurs (provided that the part of the context stated in the description itself does occur). To explicate a state is to produce an explicit description of it without having access to such a description, at the same level of abstraction, during the explication process. A process explicates a state when its output, but not its input, is an explicit description of that state. Similarly, a process explicates a process when its output, but not its input, is an explicit description of that process. In the sequel, I often abbreviate "explication of a state or process" to "explication of a situation."

Examples of explications are:

When we try to describe an event (a sequence of two or more states) to someone who has not been present at the scene of the event, we have to explicate the event. Usually, we succeed in this, though not perhaps to the extent that the group of people able to apply the result of our explication is arbitrary. There is usually still a lot of cultural background hidden in the description, as becomes apparent when we communicate the description to a member of another culture or subculture (or even of another age group).

In mathematics, finding a proof is explicating the logical relation between two or more statements. It is explicating the state of an abstract world to a very large extent. Everybody, irrespective of culture, race etc. is able to understand the resulting description, though not everybody may be able to master the background required to understand the description. But the background is fully explicated and thus available to anyone.

In empirical scientific research, scientists start with a rather vague idea about a process and see whether that idea can be put in a context-free, repeatable form communicable to other members of the scientific community. As Jefferson remarks, scientists "begin without bothering their heads about rigid definitions of what they are doing. The so-called Laws of Science had generally no very tidy beginnings." (p. 4616)

Teachers have (among other things) the task of communicating explicit descriptions to pupils, and face the explication task of finding out why a pupil does not understand an explicit description, i.e. cannot apply it. This is a very difficult task requiring insight into the background of the pupil.

In a court of law, eye-witnesses have to explicate states and processes they observed, and the result of explications by different persons are used to construct an explicit description which can be compared to existing explicit descriptions of previous cases. Usually, it is only possible to a limited degree to describe the case as if it were a machine, and a lot of personal, unexplained background is needed to be able to judge similarities and dissimilarities.

In these examples it is difficult if not impossible to produce a totally explicit description, and we often succeed only to a limited extent in explicating a state or process. Apparently, there are degrees of explicitness. A description is more explicit the more we succeed in describing a repeatable situation in a context-independent way, and in a way communicable to a larger group of people. We cannot therefore say that the ability to produce totally explicit descriptions is part of human intelligence. But would we attribute human intelligence to a being who could not explicate at all? If we would not, the ability to explicate is an essential part of human intelligence. To answer this question, we must first look at examples of explication by machine.

First of all, a human being who is a library member can be described at a certain level of abstraction as a machine, so there exist machines which can explicate. But obviously no explication process is described by that description, so the explicative ability of the human being is not described either. The library member machine cannot explicate at all.

Take, as another example, a machine of which we may want to say that it explicates. We equip a computer with a video camera and program it so that it can produce the description "this is a white ball" when shown a white ball. This is not a trivial task to program, especially when shadows of neighboring objects and an uneven background are allowed, but in principle it
can be done. We can write the program in such a way that during its computation it does not have access to an explicit description of the situation, so following the definition of explication given above, we should say that the machine explicates the situation.

But somehow, this does not feel right. The computer did not *explicate*, it performed a *computation*. But what is the difference? The machine went through a repeatable, context-independent process of which we possess a communicable description. That description is its computer program. The programmer *did* explicate the situation in all dreadful detail for the computer and embedded the description he or she came up with in a computer program. The difference between what the programmer did and what the computer did is that the situations of which the program can produce explicit descriptions are precisely circumscribed by the program text, while the range of situations of which the programmer can produce explicit descriptions is not precisely circumscribed. Present another situation to the computer, and all appearance of explication disappears: It cannot produce an output when its input is not an element of the set of valid inputs. But present a different situation to the programmer, and he or she can produce a description of it.

This account can be generalized by observing that the class of inputs to a machine must be explicitly described, because the machine itself is explicitly described. If the machine can produce explicit descriptions, the range of situations it can produce explicit descriptions of is therefore bounded by an explicit description. This contrasts with the explicative ability, however meagre, of human beings. The range of situations which any being with human intelligence can produce explicit descriptions of is not bounded by an explicit description. Any human being has the *ability* to produce explicit descriptions of a set of states and processes which hasn’t been circumscribed in advance, i.e. which hasn’t been explicitly described by a finite and constant description. We would not attribute human intelligence to a being for which we could precisely spell out which situations it can and which situations it cannot produce explicit description of.

The contrast between the presence or absence of a description of the situations which a system can produce explicit descriptions of is a part of the argument against the possibility of machine explication in the next section. Here, I want to emphasise that we do not attribute human intelligence to a being which does not have the *ability* to explicate a range of situations of which we cannot indicate the boundaries in advance. Moreover, I want to point out that the presence or absence of such an explicit description is a yes-or-no matter. In the case of machines, the description of this boundary is, by definition, totally explicit, and between total explicitness and more or less describability there is a large difference. Since the ability to explicate is bound to the presence or absence of such a totally explicit description, explicative ability is a yes-or-no matter as well. There is no such thing as possessing a "little bit" of this ability. In other words, if possession of this ability is (at least one of) the distinguishing characteristics between human intelligence and machine intelligence, then the difference between these two types of intelligence is not gradual, but discrete.

This discrete jump from machine to human intelligence has nothing to do with the ability to produce totally or partly explicit descriptions. If a machine produces explicit descriptions, it is totally explicit; if a human being produces an explicit description, it is seldom totally explicit. But the ability to produce explicit descriptions of a situation which are more or less explicit differs from the ability to explicate in a principle unbounded range of situations. It is the last one which I regard as typical for human intelligence and of which I maintain that there is a sharp distinction between the presence or absence of a bound on the explicable situations.

We cannot dodge the distinction either by a retreat to a metalevel. It might be maintained, for example, that the set of situations explicable by a human being is describable, in principle, by a fuzzy function. The function has value 0 for members definitively outside the set and is 1 for situations definitively inside the set, and has a value between 1 and 0 for members which are "more or less" in the set. The graceful degradation of the ability to explicate a situation would then be modeled by this function. But this shifts the discussion from the question whether the set of explicable situations is crisp to the question whether this function is crisp, i.e. whether for any pair \((s, n)\) where \(s\) represents a situation and \(n\) is a number between 0 and 1, we know
definitively that $f(s) = n$. If the function is not crisp, then it might be modelled in its turn by a fuzzy function, etc. But if the description of explicative ability is to remain finite, we must stop at some level, and we can resume the discussion at that level. The question to be answered by that discussion is whether the range of explicable situations is describable by an explicit description. Constructing this description out of a finite number of functions standing in certain relations to each other does not change the matter.

A similar answer can be given to the claim that we may be able to build a learning machine. If that artifact is to be called a machine, at some level of abstraction it has an explicit description. That description may describe a learning algorithm, or it may describe a way to construct a learning algorithm, etc. My claim with respect to all these alternatives remains the same, viz. because the machine is explicitly described, the range of situations it can produce explicit descriptions of is explicitly described, and therefore it does not have the ability to produce explicit descriptions of a range of situations which is not explicitly described in advance. I return briefly to the question of learning machines in the discussion of Turing (section 3.1).

Keeping the range of situations to be explicated open and unspecified is reminiscent of the following remark by Gunderson:

For if the case where the machine X-es is really the same and not just vaguely analogous to the case where man X-es, then we should be safe in making further assumptions about the machine's general capabilities and performances, just as in the case where we know that a man can do X and must thus be able to do a number of other things as well. (Gunderson [1985], p. 48)

This is Descartes' argument against the possibility that animals can think transposed to the domain of machines. Descartes argued that from the fact that animals can do one thing better than we can it does not follow that they can think. If an animal thinks as we do merely because it surpasses us in one capability, Descartes argues, then it should surpass us in any capability, so it should surpass us in thinking. Without adopting the view that animals have only a limited set of behaviors, we can agree with this argument in the domain of machines.

My account of explication leaves out all persons with below-normal linguistic ability. My argument does not imply anything either way about these persons. It does not imply that they are machines, nor that they are not machines. I do think that an argument using explication is a first step to a more comprehensive argument presupposing no linguistic ability. We would then need a concept like intentionality or Dreyfus' "zeroing in on the relevant aspects of the situation." But such an argument would use a more difficult concept than explication and would therefore be less explicit and more controversial. Moreover, using explication as a first step increases our understanding of the issue.

As a final caveat against the statement that human beings can explicate one can remark that what truly distinguishes human beings from machines is not that they can explicate, but that they can understand descriptions which are not explicit. This may be so, but some people can then easily raise the question whether someone really understands such a description. More importantly, one would then have to argue that the process of understanding descriptions which are not explicit cannot be explicitly describable. Such an argument would be more comprehensive than the current argument if it can be shown that people with below-normal linguistic ability and non-linguistic animals have the capability to understand non-explicit descriptions. An argument along these lines would be one of those improvements mentioned above, which are more difficult and more controversial.

2.4. Explication by machine

In section 2.3, the basic argument against the existence of machines with explicative ability — or of explicit descriptions of the ability to explicate, which comes down to the same thing — was already presented. To possess the ability to explicate is to be able to explicate a range of situations which is not described explicitly in advance; the range of situations which any machine
can produce explicit descriptions of is (explicitly) described in advance; so machines cannot have the ability to explicate.

This argument has some superficial similarity to the argument for the unsolvability of the halting problem. The halting problem for a particular computer program and input is to prove that that program fed with that input will terminate, i.e. yield an output. The halting problem cannot be solved uniformly, i.e. there is no computer program which can solve the halting problem for an arbitrary program and input. Such a program would take as its input 1. an arbitrary program $P$ and 2. an input $I$ to $P$, and produce as output Yes if $P$ terminates with input $I$ and No if it does not terminate with $I$. The proof of the nonexistence of such a program relies on a contradiction which emerges when such a program is given the problem whether it itself will terminate when it is given the problem whether it will terminate.\(^4\)

The general problem of showing that a program terminates for a class of inputs is not mechanically solvable by a program. Similarly, there is no general explicit description of the ability to explicate. The argument is roughly similar, because here too, a contradiction emerges when the concept of explicit description is applied to the process of explication.

But there are also dissimilarities. I mention three of them. First of all, the argument for the impossibility of the explicit description of the ability to explicate is very informal, while the proof of the unsolvability of the halting problem is formal. It could not be otherwise, for a formal proof of the impossibility to describe explicitly the ability to explicate would be self-contradictory. Such a proof would have to rely on an explicit description of the explicative ability in order for the argument that such an explicit description does not exist to be called a proof. What can be done at most is to show that the assumption that such a description exists is counterintuitive. This is what has been done, the intuition in question being that the range of situations which can be explicated is not bounded by an explicit description.

This leads to the second dissimilarity with the unsolvability proof of the halting problem. The unsolvability of the halting problem is proved using the self-referential nature of a solution to that problem. Self-referential descriptions are not necessarily contradictory just because they are self-referential, but some are, and an explicit description of a uniform procedure to solve the halting problem is a case in point. But the self-referential nature of an explicit description of explicative ability is not contradictory. Mind-boggling, perhaps, but not logically impossible.

To see this, consider the fact that given an explicit description of the ability to explicate, any explication process must be an instance of (at least) that description. Explicit descriptions are the measuring stick by which we determine whether two processes instances are occurrences of the same process. But the assumption that any explication process is an instance of that description implies that future explications are instances of that description as well. That in turn implies that we know the future to some extent, i.e. we know how explication processes will behave, just as we know how bodies with non-zero mass will behave under the influence of gravity. But in the case of explication, this consequence is much more far-reaching than in the case of the processes studied in physics. Most of the future processes have yet to be explicated. Because we may explicate in the future just about any situation for any purpose, possessing an explicit description of how these explication processes will evolve under the influence of various environmental conditions is almost tantamount to knowing in advance what will be relevant in those situations for those purposes. But even these purposes must be explicated, and those explications should satisfy the explicit description of explication as well. If those purposes in their turn are ever explicated, the argument iterates by observing that these explications are instances of the description as well, etc. And, of course, the explicit description of all this is also the result of an explication, which must be an instance of the description it produces as well. None of these consequences of the assumption that there is an explicit description of the ability to explicate are logically contradictory to each other or to the assumption, but they sure are mind-boggling.

4. A simple proof of this theorem is given in Rogers [1967], p. 24.
The third and final difference with the unsolvability of the halting problem concerns the ability to explicitly describe individual solutions to the problem in question. In the case of the halting problem, all that has been shown is that it is not uniformly solvable. It is unsolvable by one explicit description (or a finite conjunction of explicit descriptions): But in particular cases, mechanical proof of termination may well be possible and, in fact, in many cases such proofs have been found. The unsolvability of the halting problem may be construed as saying that to find such a proof, human creativity is needed. (The termination proof is not mechanical in the sense that it is found mechanically, it is mechanical in the sense that it can be verified mechanically once it is found.) 5

By contrast, in the case of explication, there are arguments why individual cases of explication cannot be explicitly described either. There are three arguments for this claim, which follow from the demands of communicability of explicit process descriptions and from the context-independence and repeatability of the processes they describe. If valid, these arguments provide extra reason to believe that a computer producing explicit descriptions of white balls is not explicating at all.

First, communicability. If an individual explication process is explicitly described, then it has an explicitly described initial state. Now, if I explicate something and try to describe the initial state of the explication process explicitly in order to communicate it to somebody, that state seems to be the most elusive of states to describe. At the start of an explication process, it is not yet clear what aspects of the situation to be explicated are relevant. Using the language of state spaces, to try to describe the initial state of an explication process is to try to describe a position

5. Myhill [1952] argues that the psychological significance of Church's theorem is that our finitude compels us to be creative. Church's theorem is that any non-trivial axiomatic theory, such as Euclidean geometry or number theory, is undecidable, i.e. there is no uniform, mechanical way to decide of a given formula whether or not it is a theorem of that theory. To be a theorem of an axiomatic theory is to follow from the axioms of the theory (and those of logic) by applying inference rules. There is a mechanical way to generate all theorems of an axiomatic theory, but the problem is that there is no mechanical way to generate all nontheorems of the theory. If a given formula is provable, it will be generated, but if it has not yet been generated, it may be because it just hasn't turned up yet or because it is a nontheorem. Since the list of theorems of a non-trivial theory is infinite, we may not be able to reach a decision. Note that this problem does not arise if we could mechanically generate all the nontheorems as well. In that case, we could turn on two machines to generate theorems and nontheorems, respectively, and be sure that the formula will turn up in one of the two lists.

Myhill's argument is that we are finite in the same way as machines are finite, so that we must generate the theorems of a theory in some order. This compels us to be creative, for

"... we have commonly at every step a wide choice of the succeeding step. This choice is made by us, keeping in mind the end as well as the initial premisses or axioms of the deduction, in accordance with principles which seem in some sense to be radically incapable of systematization. This is the locus of genius." (Myhill [1952], p. 176).

Myhill's argument is weak where he compares man's finitude with that of a machine but observes creativity at work in the generation of theorems in the case of man only and not in the case of machines. He has not shown why man and machine are analogous to each other in the case of finitude but not in the case of creativity.

My own view is that the analogy fails for finitude as well as for creativity. We are not able to oversee even a small portion of the consequences of a non-trivial set of axioms such as those of Peano arithmetic by looking at the axioms alone. But we do find some interesting consequences, given some time and resources to manipulate formulas, e.g. pencil and paper. The analogy fails here because a machine typically comes up with many more consequences in the same amount of time, but comes up indiscriminately with relevant as well as irrelevant ones. Our finitude consists best of our ability to deduce a small number of highly relevant consequences and at worst of our ability to deduce a small number of irrelevant consequences. In the case of man, there is a continuum between these two; in the case of machines, there is merely the ability to exercise the last type of finitude at high speed, i.e. produce theorems at high speed (some of which are recognized by us as being relevant. The difference between a continuum and a point is sufficiently great to say that the analogy between the finitude of man and machine breaks down. Note that this analysis agrees with the conclusion of this chapter, which is that a process which recognizes relevancy cannot always be said to be in a state and is therefore not explicitly describable.
in a state space while the subject who carries out the explication does not know yet at which position he is. At the start of explication, it is not yet clear even what the possible problem states are. At the start of explication, there is no state space. In a very real sense, there is nothing to communicate about it. The claim that there is an explicit description of the initial state of an explication process is counterintuitive because the start of an explication process is of such a nature that the concept of state is not yet applicable.

The second argument concerns the context-independence of explicitly described processes. Context-independence implies that the effects of every possible relevant input on the explication process is explicitly described. Explicit descriptions mention every relevant aspect if the environment, and what is not mentioned is not relevant. But one peculiarity of explication is that just about anything can be relevant for it. People have found explicit descriptions when staring in the fire and dozing off, or while taking a walk, or when entering a bus. We often find an explicit description of something while thinking of something totally different or doing some physical work unrelated to the problem, or when hardly thinking at all, as in the moments just before we fall asleep. Assuming that we can catch all this in a context-free description, i.e. one mentioning all and only the relevant aspects of the environment, is counterintuitive.

The third and final argument concerns repeatability. If an explication process is explicitly described, it is repeatable. That is, if the initial state and the ensuing sequence of inputs are reproduced or simply reoccur accidentally, then the explicit description predicts that the same process must ensue, producing the same outputs. Now, how does one repeat an explication process? One single person cannot repeat an explication process. Explication is irreversible (barring loss of memory, in which case one can doubt if we are still talking about the same person).

But let us write this as a pragmatical difficulty and perform a thought experiment. We place two persons (either two numerically different persons or one person at two different times, who the second time has forgotten everything about the earlier explication) and place them in the same situation. Then more often than not they explicate the situation differently. But then in order to be applicable to both explication processes, the explicit description must account for these differences. And just as before, just about anything about the two persons and their histories and background can be relevant to account for the differences in the explication processes. There seems to be no limit to what should be accounted for in the explicit description of the explication process. Of course, thought experiments about identical twins growing up in identical circumstances on identical twin planets in identical but separate regions of the universe are out of place here, since two objects can be identical only with respect to an explicit description, and the existence of such a description is what is precisely being questioned here.

To sum up, in addition to the ability to explicate not being explicitly describable, individual explication processes are not explicitly describable either. The first claim depends upon the absence of an explicit bound on the range of situations to be explicated, while the three arguments for second claim all depend upon the observation that just about anything about the person who does the explicating and the situation to be explicated can be relevant. The two claims may be combined in one single slogan. Relevance is not explicitly describable. The statement that a machine cannot explicate then becomes the statement that what is relevant about its own history or the environment is, for a machine, precisely circumscribed.

This ties in well with the concept of state, which is the essence of the concept of state evolution function and thus of the concept of a machine. A state is a precise memory of the history of the system which contains all information required to be able to describe every possible future behavior in every possible evolution of the environment in which the machine is placed. To say what the state evolution function of a system is and in which state it now is in is to state exactly what is relevant about its past and about its environment.

Conversely, to say that explication and explicative ability cannot be explicitly described, is to say that these processes are of a kind to which the concept of state is not applicable. And to the extent that these processes are essential for human intelligence, to that extent human
intelligence is a system to which the state concept is not applicable.

This undercuts any scientific effort to describe human intelligence, for scientific research is the search for state evolution functions. This does not imply that psychology as a science is impossible, for psychology studies many more processes apart from explication. But it does imply that for any psychological process of which explication is an essential part, or for which a similar argument can be given, no description can be given which takes the form of a state evolution function. Descriptions of it will deviate fundamentally from those we are accustomed to in physics. ⁶

2.5. Artificial intelligence and machine intelligence

From my argument it cannot be concluded that no artificial intelligence can be constructed. To see this, let me define an artifact as any entity constructed by man. Some machines are artifacts, like radios, TV’s, cars, and computers, but other machines are not artifacts, like the solar system, a proton, or a falling stone. There are also artifacts which are not machines under an interesting description, like paintings, novels, chairs and houses. ⁷

Following this definition, human beings are artifacts as well, because they are created by human beings. Viewed this way, the goal of AI research is to find another way to create an artifact which can think. The argument in this essay does not attempt to show that this is impossible, but that if there is a second way, then whatever is produced in that way is not a machine, i.e. we won’t have an explicit description of how it works. Or, more precisely, at the level at which we can describe it explicitly, we can’t describe its explication behavior. ⁸

---


7. There are some interesting borderline cases like cattle which are carefully bred in agro-industry and plants which are grown in totally artificial conditions, but these do not affect the argument which follows.

8. This way of viewing AI removes the excitement of boldly doing something totally unprecedented from AI research. Man creates man, and at a alarming rate at that. Another question now crops up: Why would we want to find another way of doing this? Good engineering research always has some kind of useful behavior in mind (see section 3.5). Unless an AI is to be viewed as a work of art, in which case questions of esthetics instead of usefulness should be asked, we should be able to discuss the use of AI’s just as we can discuss the use of cars, airplanes and radios.

This points to a second cluster of questions we can ask. Is it ethical to use an AI as if it were a tool? Would an AI be able to suffer? Is a being which cannot suffer intelligent? Would it be afraid to die? Can we stop its production at any moment? Can we kill it? Would it fall in love? Are the makers responsible for their AI’s and if so, are they just as responsible for the AI’s they created as parents are for their children? Is there another responsibility, connected to the fact that the makers created a type of being and are thus more or less in the position of God to Adam? Shelley’s story of Frankenstein and his creature (not the caricature presented in popular movies) considers some of these questions in a 19-th century gothic setting, and the movie Blade Runner treats them in a futuristic setting. A recent article in AI Magazine (LaChat [1986]) discusses some of the ethical problems of artificial intelligence. In a response, (Wieringa [1986]) I point out some ethical paradoxes of AI’s.
3.1. Turing

Turing [1950] supports his claim that the TT is a test for thinking by refuting nine objections to the claim that machines can think and giving one positive argument for the possibility of machine intelligence. This is a rather curious procedure: Suppose that it is true that the nine arguments attacked by Turing are wrong, then he still depends upon the plausibility of his one positive argument to lend credibility to the claim that machines can think. If nine arguments against the possibility of MI are wrong, then there may be a tenth argument which is correct. To show that such a tenth argument does not exist, it must be shown that machines can, in fact, be intelligent. But suppose that his positive argument is correct. Then machines can, in principle, be intelligent. From that we cannot draw the further conclusion that the TT is a test for thinking. To show that the TT tests the right thing, we need an independent argument.

The reason why Turing apparently thinks that he has defended the thesis that the TT is a test for thinking lies in the shift of attention from what thinking is to how we know that someone thinks. This has been discussed in section 2.2. Here, I want to discuss his positive argument for the possibility of MI and the three of his nine refutations which are incompatible with my argument.

His only positive argument is that some day, we may be able to construct a learning machine. This appeal to learning machines is perhaps a response to Jefferson's remark that

*It is not enough, therefore, to build a machine that could use words (if that were possible), it would have to be able to create concepts and to find for itself suitable words in which to express additions to knowledge that it brought about.* (Jefferson [1949], p. 1110, cited in Turing [1950], p. 17. Italics in the original.)

Jefferson made this remark to deny the possibility of thinking machines, and Turing's response is that it may well be possible to construct such a machine. My response to this is that some day, we may be able to find a second way to construct an intelligent artifact, but that Turing has not given any argument that the artifact will be a machine. In fact, the procedure he proposes, initializing an artifact with the state of mind of a human baby at birth and subjecting it to the same environmental influences as a baby, raises some conceptual problems. What, exactly, are we to understand by "initializing" in this case? And what is the "same state" and the "same environment?" We can talk about the same state and same environment only with respect to an explicit description, which then should mention all relevant aspects which are required for an intelligent end-product. It is imaginable that we reproduce living tissue in artificial circumstances, implant it in an artificial womb and then raise it like a human being. The result may well be as intelligent as any human being is -but then again, it may not. But there is no reason to suppose that we should possess an explicit description of it in order to go through the creation process, so there is no reason to suppose that it would be a machine. The argument of chapter 2 strongly suggests that it would not be a machine.

Three of Turing's nine refutations are incompatible with my thesis. I discuss each of these in turn.

My argument belongs to a group of arguments which have the form "machines cannot do X," with X = "fall in love," or "be creative," "enjoy strawberries," etc. Turing refutes these arguments by noting 1. that they are usually not argued at all and 2. that they are based on limited and outdated experience with machines (Turing [1950], p. 18). This experience is contingent, Turing says, and cannot serve as a basis for the statement that machines in principle

---

1. This draws attention again to the ethical problems mentioned in footnote 8 of chapter 2.
cannot do X. But this refutation cannot apply to my objection that machines cannot explicate, for 1. that objection is argued, and 2. it is motivated by an analysis of the concept of a machine as it is used in automata theory and systems theory.

A second objection against the possibility of MI discussed by Turing comes from Lady Lovelace, who said of the world's first proto-computer conceived by Babbage that it "has no pretensions to originate anything. It can do whatever we know how to order it to perform." (quoted in Turing [1950], p. 20/21). In so far as to explicate something is to originate something, Turing's reply to this argument is also directed at my argument and must therefore be countered. Turing's reply is that there may be machines which only do what we know how to order it to do, but which are still intelligent. For example, he says, we may subject an artifact to an education. Or, as another example, if there is an intelligent discrete-state machine, we can program a computer to behave like it.

My answer to the first reply has been given above. Turing has not shown that an educated artifact would be a machine at an interesting level of description. The second reply by Turing is hardly a refutation, for the existence of an intelligent discrete-state machine borrows its credibility completely from science fiction and not from any plausibility argument or scientific fact. Saying that it could exist without supporting it by argument is wishful thinking and hardly invalidates my argument.

As an aside, we may note that Lady Lovelace's objection does contain a possible confusion regarding creativity and the production of new, unexpected outputs. Systems are machines under a description. Their behavior is fixed by their description and in that sense, they can only do what "they are told to do." This "telling them what to do" is only metaphorical for in the case of natural machines like the solar system, the rule for constructing explicit descriptions of them is that the natural system "tells us" what an explicit description at the level of abstraction we are interested in should be. Whenever there is a significant mismatch between description and behavior, we adapt the description, not the behavior. In the case of artificial machines, on the other hand, we make the system behave in the way that the description prescribes. In case of a significant mismatch between description and behavior, we adapt the behavior, not the description.

In both cases, behavior is fixed by a description and we know the description. But if the description is complex, the system can behave in a quite unexpected and in that sense novel way. For example, an analog computer solving a differential equation which is not analytically solvable (i.e. cannot be solved by manipulating the symbols in the equation) tells us something new, viz. the solution(s) of the equation in this particular situation. The differential equation is the explicit description of the machine, so the machine does what we tell it to do, but the description is too complex for us to know fully what we are telling the machine to do. But doing things we were not able to foresee is not sufficient evidence for creative ability. Creativity is not the same as unexpected behavior, and both Lady Lovelace and Turing are very close to confusing these two. For example, Turing says that

it is perhaps worth remarking that the appreciation of something as surprising requires as much of a "creative mental act" whether the surprising event originates from man, a book, a machine or anything else. (Turing [1950], p. 22)

An unexpected act is creative only if it follows in some sense from the context of the act. It does not need to follow logically from the context, but it must have some connection with it. For example, it must be applicable to the the problem at hand, or unexpected structures of the situation must become visible, or a new viewpoint on the state of affairs must be established. Doing something unexpected is not an act of creation if it is totally unconnected to the

3. It also accords with systems description in modern science, see chapter 1. footnote 2.
environment. Random number generators are not creative.4

A third objection against the possibility of MI discussed by Turing is reconstructed by him as follows.

"If each man had a definite set of rules of conduct by which he regulated his life he would be no better than a machine. But there are no such rules, so men cannot be machines." (Turing [1950], p. 23)

The rules mentioned in this argument are like traffic rules. They are explicit descriptions which we observe, in the sense of "seeing" as well as "following." But as Ryle [1949] (p. 28 ff.) argued, not only are these rules of everyday conduct not known to us and therefore not observed by us, their existence would be contradictory as well. If the principle that man acts by observing rules were true, then to apply such a rule we would need a rule to guide us in the application, but for the application of that rule we would require a still higher level rule, etc. so that we would never come down to acting at all.

But from the fact that we do not and cannot observe such rules we cannot conclude that we are not machines. Machines do not follow such rules either. The solar system does not observe the explicit description we gave it, and neither does my radio. If we rephrase the objection to correct for this possible confusion, we get

"If each man is explicitly described, he would be no better than a machine. But there is no such description, so he is not a machine."

Put like this, it is obviously a non sequitur. The only way to find such descriptions, Turing says, is scientific observation. There are no circumstances under which we could say, "We have searched enough. There are no such [explicit descriptions]" (ibid. p. 23). But Turing is only partly correct in this. The correct observation to make is that the only way to find explicit descriptions of natural systems is by scientific observation, but that scientific observation cannot show that there are no such descriptions. But conceptual analysis can show that there are no such descriptions. For example, mathematics has shown quite convincingly, without any empirical search, that there is no square circle. On the other hand, it cannot provide us with descriptions of squares and circles occurring withing nature. As a less trivial example, Church has shown, without empirical research, that there is no algorithm for deciding whether a predicate logic formula is a theorem in any non-trivial axiomatic theory, and Gdel has shown that it is impossible to axiomatize Peano arithmetic in such a way that all true number-theoretic statements are provable. In general, all mathematical results can be viewed as impossibility claims in this way: If it has been proven that \( A \Rightarrow B \), then it is impossible to find a system in nature to which both \( A \) and \( \neg B \) apply.

I conclude that Turing's one argument in support of the possibility of MI is weak, and that three of his nine attempts at refuting objections to the possibility of MI fail. In particular, my thesis is not refuted by his arguments. On the other hand, my argument strengthens the position he wanted to refute, as is argued in the next section.

---

4. Random number generators raise conceptual problems of their own. It is very hard to describe explicitly how an object is unrelated to its environment.

5. A possible exception to this statement is Feynman's conceptual argument for the law of conservation of momentum, starting from symmetry principles. (Feynman et al. [1963], section 10-3) This does not invalidate my argument, however.
3.2. Jefferson

The Turing test is a response to Jefferson's [1949] remark that

Not until a machine can write a sonnet or compose a concerto because of thoughts and emotions felt, and not by chance fall of symbols, could we agree that machine equals brain —that is, not only write it but know that it had written it. (Jefferson [1949], p. 1110)

Turing's response is that such a machine would think because it would pass the TT. From the quotation it would seem that Jefferson should agree with this, for his statement is as much as a proposal for a TT. A machine passing a TT would have to be able to compose a sonnet and discourse about it. However, a few paragraphs earlier Jefferson says that explication is a prerequisite for conceptual thinking.

"[We] find ourselves woefully lacking in ability to describe our percepts. The variety of the visual and general perceptual scene alone is too great for those frail instruments -words- and it is because of this that literature flourishes. It is almost boring to repeat that it is because he has a vocabulary that man's intellectual progress has been made -by the day-by-day record of how far he has gone in his pilgrimage towards finite knowledge, that journey without an end. (p. 1109-1110)

To say that our descriptions cannot express the variety of events around us and that the knowledge expressed in those descriptions is finite, is one step removed from the statement that the explication process, which crosses the bridge from that infinite variety to the finite knowledge of it, is not describable in an explicit -finite- description. The argument in chapters 1 and 2 can be construed as an explication of Jefferson's remark, which strengthens it by giving reasons why machines cannot explicate.

3.3. Semantic information processing

It is sometimes maintained that AI programs specify not just any information processes, but semantic information processes (Dennett [1981b], Haugeland [1981]). By this is meant that they have representational content, i.e. are about something. Added to the assumption that "intelligent beings are semantic engines -in other words, automatic formal systems with interpretations under which they consistently make sense" (Haugeland [1981], p.31), this leads to the hypothesis that

people and intelligent computers (if and when there are any) turn out to be merely different manifestations of the same underlying phenomenon. (ibid.)

If computers and people are merely different manifestations of the same underlying phenomenon, then there are machines which can explicate. So either this hypothesis is correct and my argument is wrong or the hypothesis is wrong and my argument correct. It is not difficult to show that Haugeland's hypothesis is wrong.

First of all, Haugeland argues that the following two premisses may turn out to be true:

1. intelligent beings are semantic engines
2. AI programs specify semantic engines

But from these two premisses we can conclude only that the set of semantic engines includes the set of intelligent beings as well as the set of systems specified by AI programs. Nothing follows about the intersection of the set of intelligent beings with that of the set of systems specified by AI programs. In particular, it does not follow that there are AI programs which are intelligent.

But though the argument may be wrong, maybe it can be fixed as follows.
(1') semantic engines are intelligent beings
(2') AI programs are semantic engines
(3') AI programs are intelligent beings.

The claim that human beings and AI programs are species of the same genus can be explicated as

(4') human beings are semantic engines.

Together with (1') we can then conclude that human beings are intelligent beings for the same reason as AI programs are intelligent beings. (Actually, we need a stronger claim than (1') to support this conclusion, viz. that only semantic engines are intelligent beings. Otherwise, there might be other reasons, besides being semantic engines, why human beings are intelligent. In what follows, I will refute (1'), so that consideration of this stronger claim is not necessary.)

There are three claims to be substantiated here, (1'), (2') and (4'). These arguments depend crucially upon the concept of representational content, so I will explicate that first.

Human beings have mental states, which are characterized (following Searle [1983], but I think what follows will be accepted by all parties in the game) by their representational content and their psychological mode. For example, if I want you to leave the room, the representational content of the mental state I am in is that you will leave the room, and the psychological mode is that of wanting. If I believe it is raining, the representational content of the state I am in is that it is raining, and the psychological mode is that of believing. Now, this is very near common sense, though a detailed treatment will encounter a number of problems which have no easy solutions (see Searle [1983]). But armed with these simple examples, we can explicate the claim that semantic engines are intelligent as the claim that semantic engines have states with representational content and psychological mode. In discussions of this claim, only the first half has receive attention, so I will concentrate upon that part of it only. Claims (1')-(4') are accordingly reduced to

(1'') semantic engines have representational content.
(2'') AI programs are semantic engines
(3'') AI programs have representational content.

(4') human beings are semantic engines.

Only (1'') and (3'') differ from (1') and (3'); the other two statements are unchanged.

To understand the arguments given in favor of (1'') and (2''), it is useful to see what is involved in (3''). An AI program is at least a simulation of cognitive processes (and according to (2''), it is more than that). There are therefore two modes of representing involved in (3''). First, an AI program which simulates the belief that it is raining, represents that belief in the same way as a simulation of a thunderstorm can be in a state which represents a state of the thunderstorm. This way of representing is identical with the representing in any simulation program. Clearly, the existence of this representation in a medium is no reason to assume that the medium (automated or not) shares in any way properties with what is represented. It has been remarked frequently that just as a simulation of a thunderstorm is not wet, a simulation of a belief has no representational content. Figure 1 makes the situation clear. This introduces the second mode of representing in the particular case of cognitive simulation. The represented belief has representational content. This is a crucial difference with the case of the thunderstorm, whose states have no representational content. What is needed in the case of cognitive simulation, is an argument why the presence of representational content in the simulated states warrants the claim that the simulating states have representational content as well. Whatever this argument is, it should be applicable to cognitive simulations and not to physical simulations. We can use the simulation of a thunderstorm as a touch-stone to check this requirement.

Now, (3'') says that an AI program has states with representational content and gives (1'')
<table>
<thead>
<tr>
<th>state in cognitive simulation of belief</th>
<th>represents</th>
<th>belief state</th>
<th>directed at</th>
<th>object of belief</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>has</td>
<td>representation content</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1

and (2'') as argument. Let us look at the support which Haugeland [1981] gives to (1'') and (2'') and see whether it applies to cognitive simulations and not to simulations of thunderstorms.

Haugeland defines a semantic engine as an automatic formal system with an interpretation such that the syntactical rules of the formal system are truth-preserving. A syntactical process is defined as a process in which formal tokens (e.g., marks on paper) are moved around according to explicit rules. Such a process is truth-preserving with respect to an interpretation iff the rules will transform tokens which are true in the interpretation into other tokens which are true in the same interpretation. Because of this truth-preservation, Haugeland says, states in a semantic engine have representational content, thus substantiating (1''). (2'') is supported by observing that AI programs are truth-preserving automatic formal systems. (4'') is not explicitly argued. I will now discuss these substantiations.

The shortest remarks can be made about the support for (2'') and (4''). I will deal first with (4''), because support for it is virtually non-existent. The only reason I can discover for believing (4'') is that human beings are in states with representational content, so they must be semantic engines. But this depends upon an inversion of (1''), viz.

Systems with representational content are semantic engines.

This is the original claim (1). No support is given for this claim, so I take it that (4'') has not been argued by Dennett and Haugeland.

The support for (2'') is, as far as I can see, correct. An AI program is (at least) a cognitive simulation, and a cognitive simulation is a formal, automatic, truth-preserving system. But I also think that (2'') is correct for our touch-stone, the simulation of a thunderstorm, so that the argument (1'')-(2'')-(3'') collapses. To see why (2'') applies to a simulation of a thunderstorm, I will change to less exotic terminology to describe the same situation.

A formal system is what is usually called an axiomatic system, a list of axioms in predicate logic, together with inference rules to derive theorems from the axioms. The axioms and inference rules are purely syntactic, i.e., they move around symbols according to their form. These symbols can be marks on paper or states in a computer memory or any other discrete set of distinct elements. The system in which these symbols are realized is usually called the implementation medium. In an automatic formal system, the implementation medium carries out the symbol manipulations automatically. Examples of automatic formal systems are TM's (of which UTM's are a subclass) and digital computers.

In predicate logic, meaning is reduced to the denotation of linguistic symbols by choosing a universe of objects and assigning objects in that universe as the denotation of formulas of the language. The precise way in which this is done is explained in any textbook on predicate logic and is irrelevant here. Important is that the axioms and theorems of an axiomatic system have a meaning with respect to an interpretation in a universe of objects. I will refer to this concept of meaning as "interpretation in the standard predicate logic sense."
In a purely syntactic system, the question of the truth of a formula does not arise. With respect to an interpretation, however, a formula can be true or false. An interpretation in which a set $S$ of formulas is true is called a model of that set. Now, suppose we have a model for the axioms of an axiomatic system, we can ask the question whether the inference rules, which are defined purely syntactically, are such that the theorems which can be derived by them are true in the model as well? An axiomatic system for which this is the case, regardless of the interpretation we choose, is called sound. In a sound axiomatic system, any model for the axioms is a model for all theorems. Soundness is what Dennett and Haugeland call truth-preservation.

The upshot of all this is that, by definition, a semantic engine is an automated, sound axiomatic system. This includes just about any simulation program. A simulation of a thunderstorm can be viewed as a an extension of the axioms of predicate logic with axioms (empirical laws) describing the behavior of thunderstorms (at a certain level of abstraction). A computerized simulation of a thunderstorm is therefore a semantic engine.

These explications put us also in a position to judge the value of (1''). The claim is that an automated, sound axiomatic system can be in states which have representational content. Applying this to our touch-stone, states in a simulation of a thunderstorm have representational content. Now this cannot be true unless we radically re-interpret the phrase "representational content." The way in which this phrase must be reinterpreted to make sense of the statements that a simulation of a thunderstorm can be in states which have representational content is that "representational content" means the same as "interpretation" in the predicate logic sense. "State X is the interpretation of symbol Y" is then synonymous with "state X is the representational content of symbol Y." The situation is, on this view, much simpler than that of figure 1 and is shown in figure 2.

```
computational state      directed at   object
                          
                          has
                          
representational content
```

Figure 2

The representational content line now coincides with the interpretation line. If this picture is correct (which I believe it isn't), then we are justified in saying that the computer believes that we believe that it is raining. Haugeland c.s. presumably have something different in mind, viz. that if the computer is in the appropriate state, it believes that it is raining, it believes it in the same way as we do when we believe that it is raining. The picture which expresses this now collapses into figure 3.

```
computational state    directed at   object

belief state           directed at   object
```

Figure 3

The difference between the two pictures is that in figure 2 the computer simulates a belief state while in figure 3 both the computational state and the belief state simulate (which is now identified with "represent," "is interpreted in") an object. In figure 3, simulation of the belief state has turned into duplication. I will not analyze this confusion here but concentrate on another
mistake. In both pictures, interpretation in the predicate logic sense is confused with representational content. Put differently, a computational state (which is a formal symbol to which an interpretation function can assign a denotation) is viewed in the same way as a belief state (which is a mental state with representational content).

But this equivocation between computational state and mental state leads to the well-known homunculus mistake. A symbol in a formal system is given a meaning by an agent external to it. We are allowed to give any interpretation to a sound (and consistent) axiomatic system. These interpretations are therefore not internal to the system. As is argued above, the agent assigning the meaning is, ultimately, an intentional being, i.e. a being who can be in states with representational content. The link between the intentional agent and the interpreted formal system may be mediated by a series of automated interpretation functions, but this does not affect the argument. (The automation of interpretation functions is treated below when we discuss Pylyshyn’s standpoint). If "representational content" is re-interpreted as "interpretation" and if a belief by a human being has representational content, then a belief state is interpreted. But who assigns the interpretation to the mental state? We must postulate a homunculus who does the interpreting and for whom the belief-state has representational content. The homunculus is an intentional being which can be in states with representational content, so the argument is back at square zero: Who assigns meaning to the states which the homunculus can be in? We get an infinite regress in the argument.

Part of the reason for distinguishing representational content from interpretation in the predicate logic sense is, precisely, to avoid this infinite series of homunculi. As Searle ([1983], p. 17, p. 60) remarks, the content of a belief is not the object of that belief. Some intentional states, like the famous belief that the king of France is bald, are not about anything, but they do have representational content, i.e. that the king of France is bald. Independently of the question whether this is a good solution to the problem of intentional objects, the distinction is valid in uncontroversial cases. If I am standing outside in the rain and believe that it is raining, my belief is an intentional state directed at rain, the wet substance falling from the heavens. Because of that, the belief has the representational content that it rains. My belief is not directed at its content "that it rains" but at the rain falling on my head.

At this point, the focus of the argument is usually shifted from truth-preservation (i.e. soundness in the predicate logic sense) to automation. A semantic engine is an automated system. Fodor [1968] and Dennett [1971], [1978] admit that we do need higher homunculi, but that in an automated system these are progressively simpler and that the series stops at the point where a homunculus is simply a subroutine in a computer program (or a piece of hardware executing a subroutine). In this view, the existence of automated formal systems shows that the argument leading to the infinite series of homunculi is mistaken, and is mistaken where it says that a homunculus must be intentional in the same way as the system whose states it interprets is intentional. Computers show that symbols can be interpreted by non-intentional systems.

For example, Dennett [1971] (p. 12) and [1978] (p. 123) argues that AI research "takes out intelligence loans" by breaking down the problem of simulating intelligence into smaller problems which are postulated, for the time being, to have been solved by subprograms having, again for the time being, intentionality. This loan must be repaid later on by filling in each of these subprograms by real mechanical procedures, presumably by repeating the process, i.e. breaking the subprograms down into even smaller programs which are, for the time being, supposed to be intelligent. Similarly, Fodor [1968] argues that when we tie our shoes, there is a little man in our head which applies instructions about how to tie our shoes. The homunculus is a set of psychological faculties (p. 65), the instructions are elementary operations executed mechanically by our brain (p. 66). The intended analogy between mind/brain and computer software/hardware is clear.

How is one to criticize the position taken by Dennett and Fodor? Three moves can be made in the critique. Interestingly, the third move take us back to the concept of a state discussed in chapter 1 and argued to be inapplicable to at least one psychological process in chapter 2.
The first move is that there is no psychological theory even remotely similar to the one Dennett and Fodor envisage. But of course, this can be written off as an irrelevant detail which is contingent upon the current state of scientific knowledge. Critique of the picture must show that the account is wrong in principle. Still, we may remark that a loan must be repaid, and if then debtor waits too long with even an initial repaiment, his credentials are doubted.

But there are more principled criticisms. The second move, then, takes its clue from Ryle [1949]. If intelligence is the application of instructions, why are we not aware of applying instructions? Ryle (p. 29), and, following him, Fodor [1968] (p. 68-69); answer this by assuming that the application of instructions goes on unconsciously. They then merrily proceed to the third move in the argument. Before following them, I want to draw attention to a problem raised by this answer, which is that no reasons have been given why we would be unaware of the application of some instructions but aware of others. In general, the postulated unconscious application of rules is much faster than the conscious application of rules, so that efficiency considerations would favor total unconsciousness. This point has been raised by Dreyfus [1979] (p. 106) and will be taken up briefly in section 3.5. Here, I merely point out that no AI researcher has (to my knowledge) answered this challenge.

Now for the third and essential move. If intelligence is the application of instructions, then, just as for any application of rules, this can be done in an intelligent or stupid manner. In order to apply the rules intelligently, we need a higher order intelligence which, if it is of the same nature as the lower order intelligence, applies rules and therefore requires a still higher order intelligence, etc. (Ryle [1949], p. 28-31). This reasoning takes us back to the infinite series of homunculi.

Fodor has an answer to this. First of all, he says, the instructions for applying X are the instructions for doing X well ([1968], p. 73). Therefore, we do not get a regress by having to postulate instructions for applying instructions well. Secondly, there must be instructions, for if we can say of an activity X that there is a way to do it, then there is an answer to the question "How does one do X?" ([1968], p. 74) In other words, if it makes sense to say that there is a way to do X, there is a description of X, even though the actor may not be aware of the description and may not be able to explicate how he does X.

Now, the description applied by a homunculus must be explicit. This is so when the homunculus is a computer program, but this is also the case when it is a homunculus postulated by scientific psychology. Science searches for explicit descriptions. When a description of the how of an act has been given which is not communicable, or not applicable to different instances of the act, or does not mention all relevant data about the environment, in a scientific sense no description has been found yet. This observation allows us to conclude that there is at least one psychological process for which no finite series of progressively simpler homunculi can be postulated: Explication. There are no explicit descriptions which any of these homunculi can apply in the case of explication. So in the case of explication, the shift from a homunculus who interprets psychological states to a homunculus who applies instructions does not work. It is important to see that this shift of focus does not work, not because there is an infinite regress of homunculi, but because there is not even one homunculus. In the case of explication, the series of homunculi is empty. It makes sense to postulate a homunculus when the process it must execute is a process of which it makes sense to say that it has states. If we cannot say of a process that it is in a state, there is no explicit description of it. Explication is one of the things which an agent can do of which he cannot say how he does it and of which, contra Fodor, there is no definite answer to the question, "How does one do it?".

---

6. Fodor [1983] has turned to the opinion that higher mental processes are not explicitly describable. In other words, Fodor thinks Dennett cannot repay his intelligence loans. This is consistent with the view Fodor expressed 15 years earlier, that there must be a description of process X if it makes sense to ask how X is executed. Apparently, in his [1983] Fodor thinks that there is no way in which one can be said to execute higher mental processes, at least not in the same sense as one can say that there is a way of tying one's shoe-strings. I return to Fodor's view at the end of section 4.2.
It is good to step back at this point and review the argument. The question to be answered is whether \((2^2)\) is true, viz. whether semantic engines can be in states with representational content. Semantic engines are automated, sound axiom systems and if the question is answered affirmatively, then "representational content" is construed as synonymous to "interpretation" in the predicate logic sense. This equivocation leads in human beings to the need for a homunculus who assigns representational content (an interpretation of the symbols) to mental states. Since the homunculus must be capable of having states with representational content as well, we get an infinite regress. Fodor and Dennett seek a way out of this regress by viewing "interpretation" and therefore "having a representational content" in the case of automated systems in a procedural manner. The representational content of a mental state (and process) is interpreted by a homunculus which is simpler than the homunculus whose mental states (and processes) are implemented. Mathematical niceties about the conditions under which a decreasing series of numbers is finite apart, this is taken as an argument for the fact that the regress is not infinite. But this way out depends upon the assumption that the executed processes have explicit descriptions which can be executed, and for at least one mental process, this has been shown (not proven) not to be the case (or highly counterintuitive, very implausible). We can therefore wind up the argument by concluding that in at least one important case representational content is not the same as interpretation in the predicate logic sense. This warrants the conclusion that "representational content" and "interpretation" are not synonymous. In the simulation of a mental state, we cannot put the representational content of the mental state at the place of the question mark in figure 1. Figure 2 must therefore be wrong.

But we are not yet at the end of the resources mustered by the proponents of the idea that states in a semantic engines have representational content. Pylyshyn ([1980], p. 443 and [1984], pp. 44 ff.) argues that a state in a process has representational content if we miss generalizations about the process when we don't make the assumption that it has representational content. Since, he says, we miss generalizations about cognitive simulations when we do not assume that they have representational content, they must have representational content. I have two objections to this viewpoint. First, if I believe that it rains, the content of my mental state is that it rains. The state has this content, not because someone else can only make sense of my behavior if he attributes that content to my state, but because I have the belief that it is raining (cf. Searle [1980b], p. 453). In fact, I may show no behaviour at all to make sense of, so no generalizations need be made. And I still can have mental states with representational content when I show no behavior.

Secondly, Pylyshyn has not shown that we miss generalizations about cognitive simulation programs if we describe them in mechanistic terms. We miss generalizations if we describe them at the level of bit operations (Pylyshyn [1984], p. 27), but this is true of every computation, including simulations of thunderstorms or mental states. We can capture generalizations about a computation at any of the levels at which we can meaningfully describe a computer. To see this, let us look a bit closer at the nature of computation and simulation.

When I compute the value of an arbitrary mathematical function \( f \) for an argument \( a \) with pencil and paper, I write down a series of symbols on paper which can be interpreted in an abstract universe of mathematical objects (figure 4).

\[
\text{symbols} \quad \text{interpretation function} \quad \text{mathematical objects}
\]

Figure 4.

If I compute the pressure of a gas from its volume and temperature, the symbols (e.g. \( P \), \( V \) and \( T \)) can be interpreted in a universe which is an abstract representation of the gas. The symbols are interpreted in an abstraction of the real gas. Irrelevant details, measurement errors, random fluctuations etc. are abstracted away (figure 5). If I subsequently compute another \( P \) from another \( V \) and \( T \), I engage in a simulation of a process. I have simulated the process by
computing two states in it. A *simulation* thus consists of a time-ordered sequence of configurations like the one shown in figure 5. Note, incidentally, that the speed by which a configuration of \( P, V \) and \( T \) on the left-hand side of figure 5 is replaced by a new configuration is very much slower than the speed at which the states of the gas which they represent follow each other up. Also, to reach a new \( P, V \) and \( T \), many intermediate computational states occur which have no interpretation in the abstract model or the gas. Both these points are typical for simulations.

The *implementation medium* in these computations consists of myself and the pencil and paper I use. To be able to carry out the computation, I, as part of the implementation medium, must be able to recognize and manipulate symbols. This is interpretation in the procedural sense, which must be distinguished from interpretation in the predicate logic sense. In order not to confuse the two senses of interpretation, I will call the procedural one *execution*. We now have figure 6.

![Figure 6](image)

When we automate the computation, the implementation medium is replaced by a machine. The description of the computation is then broken down in simple steps, which may be called subroutines. To picture this, we imagine figure 6 duplicated as many times as the number of steps needed to complete the computation, and connect each picture (except the last one) with the next by the name of the computational step executed. As in the manual case, the sequence of states on the left-hand side of the figure evolves much slower than that on the right hand side, and between two computational states \( S_1 \) and \( S_2 \) which represent states of the gas, there are intervening states which have no interpretation in the gas. When we make a cross-section of the computational sequence at the left hand side of figure 6 (perpendicular to the paper), we get figure 7.

![Figure 7](image)

Each \( P_1, V_1, T_1 \) represents a state in the gas. Intervening between these states, there may be configurations of \( P, V \) and \( T \) which are meaningless as far as the simulation is concerned, but which are required in order to execute the computation in non-zero but finite time in the implementation medium. Each subroutine is an execution of a series of instructions.

Now, the subroutines in a computation are usually broken down in smaller chunks, each of which takes one state of the subroutine computation to the next state. Zooming in on *subroutine_1*, for example, we see figure 8. At this greater level of computational detail, there are more state variables than just \( P, V \) and \( T \). This is represented by adding \( X \) and \( Y \) as state variables to the computation of \( P_2, V_2, T_2 \) from \( P_1, V_1, T_1 \). Step 1 starts with \( P_1, V_1 \) and \( T_1 \) as values for \( P, V \) and \( T \) and initializes the variables \( X \) and \( Y \) to \( X_0 \) and \( Y_0 \). It delivers \( X_1, ..., T_1 \) as values of its state variables, which are given to step 2, etc. At no moment have \( X \) and \( Y \) any meaning in
terms of the abstract model of the gas, and only at the beginning and end of subroutine 1 have $P$, $V$, and $T$ a meaning.

The breaking down of steps in smaller steps is continued until elementary computer instructions are reached. We thus get a hierarchical computation structure, in which each level of the hierarchy is interpreted procedurally, or executed, by the level below. How this is in fact done is described very well in Tanenbaum [1978].

We now have a two-dimensional picture of simulation. In one dimension, shown in figure 6, symbols are interpreted in a standard predicate logic way in an abstract structure. The second dimension of simulation is a hierarchy of temporally ordered steps, each lower level of which executes the steps above. We can now turn to Pylyshyn's claim that we would miss generalizations about cognitive simulations when we do not attribute representational content to them.

From the above account it is immediately apparent that we miss generalizations about the computation at any level of abstraction. When the computation has been broken down into $n$ levels, a description at level $i$ misses generalizations which can be stated at one of the other $n-1$ levels. Therefore, to say that the description at level $i$ misses a generalization does not invalidate that description. It merely needs to be supplemented, depending upon one's purposes, by descriptions at other levels.

Secondly, the only interesting generalizations to be made about the computation from the simulation point of view are those made at the highest level in figure 6, and made in terms of the abstract model of the subject of simulation. All levels downward from $P$, $V$ and $T$ in figure 6 are irrelevant as far as the subject of study is concerned. Conversely, without changing anything in the implementation, we can reinterpret the symbols $P$, $V$ and $T$ and come up with a different simulation. At a certain level of abstraction the behavior of a hydraulic system is described by the same equations as the behavior of an electric network and a simulation of a process in a hydraulic system can therefore be reinterpreted in an electric system. The implementation medium and the subject of simulation being so independent of each other, the fact that we need certain concepts (like representational contents, for example) to describe generalizations about the abstract model of the subject says nothing about the implementation. It only says something about the abstract model (and, hopefully, about the subject itself).

I conclude therefore that Pylyshyn has not given a reason to assign a representational content to automatic formal systems. (1"), which says that semantic engines have representational content, is therefore unfounded.

As a suggestion why so much mental energy is spent on the idea that automatic formal systems have states with representational content, I offer the following. Pylyshyn, Dennett and Hauge­land (and perhaps Fodor) may have been misled by our using antropomorphic terms like "knowing," "believing," "missing," and "desiring" in our descriptions of computations. But the reason we do this is just that it is convenient, not that it is true. We have a deep-seated preference for animistic thinking which we share with other cultures (Hunnings [1972]). Descriptions of computations which do not use them are difficult to read. Given these temptations, it is perhaps good to remind oneself of a warning Jefferson gave of the danger of anthropomorphic thinking.

What I fear is that a great many airy theories will arise in the attempt to persuade us against our better judgement. We have had a hard task to dissuade man from reading qualities of human mind in animals. I see a new and great danger threatening -that of anthropomorphizing the machine. (Jefferson [1949], p. 1110)
Animistic attributions are convenient when they are used metaphorically, but we should realize that they are just colorful words which play no explanatory role in the description of the computation.

3.4. Searle

Searle [1980] describes what has since come to be known as the Chinese Room Experiment. A man sits in a blinded room and receives three pieces of paper through a hole in the wall. The first contains a story written in English, the second a story written in Chinese and the third a set of instructions, written in English, to manipulate the Chinese characters. The man is a native speaker of English but does not understand Chinese. He is not even sure that the Chinese text is written in Chinese and not in another exotic language. He is now given pieces of paper containing 1. questions in English about the story, the answers to which he has to write down and give to his interrogators outside the room and 2. questions written in Chinese, from which he has to produce new Chinese texts using the original Chinese story and the symbol-manipulating program. His interrogators outside the room know that he answers Chinese questions about a Chinese story in Chinese. But as far as the man is concerned, he is just manipulating squiggles. He gets so dexterious at this that his interrogators notice no difference in speed with which he puts out English answers and Chinese answers. With respect to the Chinese text, Searle says, the man in the room is in the same position as a computer executing a program; with respect to the English text, however, he is in the same position as any other native speaker of English.

Searle then discusses the relevance of this example to the claims made by what he calls strong AI research. The claims of strong AI research are that

1. A computer programmed in a way similar to the man in the room understands the Chinese story, and
2. That that program in some sense explains that understanding.

I follow his discussion for the first claim only.

Searle’s response to the first claim is that, obviously, the man does not understand Chinese, at least not in the way that he understands the English story. Since the man in the room is doing essentially the same as a computer, Searle concludes that the first claim of strong AI research is false. Put differently, the TT is not a test to find out whether someone understands a piece of text.

Secondly, Searle says, an underlying assumption of the TT and of AI research, which he calls the strong AI thesis is hereby refuted too. The strong AI thesis says that human intelligence is the only biological process which is implementation-independent, i.e. can be realized in different media. In the words of Hofstadter [1979], mind is "skimmable" from brain. In the following paragraphs, I sum up Searle’s discussion of counterarguments to this refutation before evaluating his own standpoint. Searle discusses four major and two minor counterarguments to this claim. The first minor argument is that we only have behavior to go on if we want to justify attribution of intelligence to people and machines alike, so that if we refuse to attribute intelligence to the one, we have to refuse to attribute it to the other. Searle’s refutation of this has been discussed above already (2.2). The point is not how we know that other people are intelligent (nor how we justify that claim) but what I am attributing when I attribute intelligence to them.

The second minor counterargument is that some day, we may be able to produce an artificial intelligence in a medium radically different from digital or analog computers. This is virtually identical to the only positive argument Turing gives in support of his claim. My answer to it

---

7. The following description deviates slightly from Searle’s description.
8. It is inconsistent with the "skimmability theses," to say the least, when after a number of contorted versions of the TC thesis, Hofstadter [1979] (p. 579) comes up with the "AI thesis: As the intelligence of machines evolves, its underlying mechanisms will gradually converge to the mechanisms underlying human intelligence."
is given in 3.1: We may be able to produce an intelligence in a novel way, but if the result is
intelligent in the way human beings are, we cannot describe its intelligence explicitly. Searle's
answer is that in the case that the medium is radically different from a digital computer, his
argument does not apply. The Chinese Room argument is directed at the claim that computers
can be intelligent, have cognitive states, merely by executing a program. When that claim is
replaced by another one, the goal of AI research is trivialized by "redefining it as whatever
artificially produces and explains cognition" (Searle [1980a], p. 298).

Implicitly, this is an admission by Searle that he has not refuted the strong AI thesis. He
admits that he has shown at most that intelligence cannot be implemented in computers. But the
thesis of implementation-independence of intelligence is stronger than that. It says that intelli-
gence may be realized in other media than the brain. If it cannot be realized in a computer,
then maybe it can be realized in yet other media.

The four major counterarguments are as follows.

1. The systems reply says that the man in the room may not understand Chinese, but the sys-
tem as a whole does. Searle answers that we may dispense with the system and let the man
internalize all the squiggle-manipulating rules. He then is the system, but still does not
understand Chinese.

2. The robot reply says that we may add sensors and effectors to the linguistic in- and output
organs of the system, so that it can act, talk, see, walk, etc. Such a robot, it is claimed,
would have mental states. Searle's answer is that we can, again, put a man in a room and
feed him with all sorts of incomprehensible instructions which he has to mindlessly exe-
cute, while outside the room and unbeknown to him we connect the streams of symbols
going in and out of the room with sensors and effectors. Then the mental states of the man
are not of the relevant type and the robot he implements does not have mental states.

3. The brain simulation reply is the ultimate TT: Simulate all events in the brain down to the
neuron level. The resulting system will then think. Searle replies that the simulation only
simulates the formal structure of the sequence of neuron firings and would lack the "causal
powers" of the brain and would therefore not think.

4. The combination reply, finally, proposes to combine all the foregoing replies into one. A
robot in the form of the brain is put in a skull and connected to a computer programmed
to simulate the firing of synapses. If we think of this as one system, it thinks. Searle's
answer is that we would attribute intentionality to it, pending some reason not to. One rea-
son not to is that we discover its program. If we know how to account for its behavior
without assuming that it has mental states, we would not attribute mental states to it. We
can instruct a man in a room to execute the program, and that man, nor the system he is
part of, would not have the relevant mental states.

The first evaluative remark to be made about these counterarguments and Searle's replies is that
it is increasingly hard to have any reliable intuitions about them. For example, if the man learns
the squiggle-manipulating program by heart and can apply it as fast as he can speak English,
what is the difference with someone who learned Chinese consciously by memorizing grammar
rules and applying them? If he learns to understand Chinese in the second case, why not in the
first? Or if the brain is duplicated in a material which behaves in the same way down to the
neuron level, what is the difference between a real brain and its simulation? If the processes are
similar in all describable aspects, why should the brain think and not its duplicate? The original
intuition from which the Chinese room experiment derives its appeal, the difference between
symbol-manipulation and human understanding, is snowed under by a shower of confusing
details.

The second remark I want to make about Searle's answers is that there is a contradiction hid-
den in them. Uncovering this contradiction will make clear why Searle's answers to the coun-
terarguments are weak as well as point out what the answer to the counterexamples should be.

The contradiction is hidden in the unanalyzed concepts of "formal program" on the one
hand and "causal powers of the brain" on the other. Searle compares the causal powers of the
brain to produce mental processes with the causal power of \( H_2O \) to produce the liquid properties of water. Just as "the liquidity of water is not to be found at the level of the individual molecule," so "the visual perception [is not] to be found at the level of the individual neuron or synapse" (Searle [1983], p.268). In general,

"the surface structure is both caused by the behaviour of micro-elements, and at the same time is realised in the system that is made up of the micro-elements" (Searle [1984], p. 21, Searle's italics).

If we look closer at this analogy, we will see, first, that it breaks down, and second, that the concepts of "caused by" and "realized in" are confused. When we remove the confusion, the contradiction alluded to emerges.

The analogy breaks down because the relation between liquidity and the microstructure of water is not of the same type as the relation between a visual experience and the microstructure of our brain. If a system is liquid, it is liquid because it can be divided in subsystems (its microstructure) whose state evolution functions and interaction work together to produce output when given certain input. We call that input/output behavior "liquidity." But if a system has a visual experience (I, looked at in the third person), its microstructure (neurons, retina etc. with their state evolution functions and interaction patterns) do not produce behavior which we, the observers of the system, call "visual experience." There is no behavior called "visual experience." The subject can report on visual experience, but this is verbal behavior, not visual behavior (whatever that may be), let alone visual experience. The microstructure of neurons, retina etc. does not produce input/output behavior which can be classified by the observers as a "visual experience"; in contrast, the microstructure of water does produce I/O behavior which can be classified by observers as liquidity. Whatever behavior is observed by external observers is not visual experience, while whatever is observed by the subject himself is not the behavior of his neurons and retina, but the objects or events which the experimenter displays to him. This difference between the relation between liquidity and the microstructure of water on the one hand and visual experience and the microstructure of the brain on the other, is obscured by Searle.

Secondly, the phrases "caused by" and "realized in" obscure a distinction which ought to be made in this context. When the liquidity of water is explained by the microstructure of water, there are two different levels of abstraction, as illustrated in figure 9.

\[
\begin{align*}
H_2O & \quad \text{explains} \quad \text{parameter ranges} \quad \text{abstract} \quad \text{liquidity}
\end{align*}
\]

Figure 9.

The microstructure of water explains why certain parameters describing the behavior of water fall in certain ranges. These parameters with these ranges are an abstract, explicit description of what in daily life goes under the name "liquidity." Now, the explication of the common concept of liquidity as certain explicitly defined parameters falling in certain ranges does not exactly coincide with the unexplicated concept of liquidity. Certain viscous substances will be called liquid in the abstract sense which no one would call liquid in the common sense of the word, so the abstraction applies to objects to which the common sense concept does not apply. Conversely, with the common sense concept goes a halo of poetic associations and memories (a guitar solo was once described to me as "wet") which are deemed irrelevant to the behavior of liquid substances and are abstracted from. So the intuitive concept applies to objects and processes to which the abstraction does not apply.

Given the mismatch between intuitive concept and abstraction, one can for any abstraction ask whether it adequately captures the part of the intuition we want to characterize. We can relegate some parts of the intuitive concept to irrelevance and we can adapt its meaning to that
of the abstraction, but intuitive it remains, and we can always ask the question how close the match between intuition and abstraction is. The fundamental problem here is that there can be no explicit proof that the two concepts coincide. Such a proof would forestall doubts about the validity of the abstraction as an abstraction of the intuitive concept. This is the problem which is central to the Turing-Church thesis: It cannot be proved, only made plausible, that the concept of a recursive function is an adequate explication of the concept of an algorithmic function.

Now, Searle's relations of "caused by" and "realized in" are represented by the edge labelled "explains" in figure 9. Or, to be more precise, the behavior characterized by the parameters in question is caused by and realized in $H_2O$. What is not caused by and realized in $H_2O$ is the liquidity which is the meaning of the intuitive concept at the right hand side of figure 9. Searle confuses the matter by identifying an intuitive concept with its abstraction.

The distinction is important in the present discussion, for it allows us to state what is wrong with the analogy between $H_2O$/liquidity and neurons/mind. Consider what the diagram should look like in the case of visual perception (figure 10).

$$\text{neurons &c. } \text{explains} \quad \text{brain behavior} \quad ???? \quad \text{visual experience}$$

Figure 10.

Brain behavior is caused by and realized in the brain, but what is the relation between brain (and retina) behavior and visual experience? Certainly not one of abstraction. And yet visual experiences must take place in the common sense world, while brain behavior takes place in the abstract world of science. A fairly standard answer in cognitive psychology is that the question marks should be replaced by a series of systems which, with respect to the behavior of the enormous heap of $10^{10}$ neurons (with $10^{800}$ potential connections) are abstractions, in such a manner that the system with the highest level of abstraction is also an abstract description of the visual experience. I will not pursue this answer but simply note that whatever is filled in for the question marks, it cannot change the fact that the visual experience to be explained takes place in the common sense world just as the intuitive concept of liquidity has a meaning in the common sense world.

The conclusion is that whatever is caused by and realized in the brain, it is not the visual experience which the subject reports on, but an abstraction of it. The corresponding conclusion in the case of water is not very disturbing, for it is we, not the water, who produce the vagueness of the everyday concept. But in the case of the visual experience the analogous conclusion is disturbing, for it is presumably our brain which produces the vagueness of common visual experience. As scientists we should therefore try something which in the case of water we do not have to undertake: Explain the relation between the vague common visual experience and the microstructure of the brain. But a glance at figure 10 (and at my argument) shows that the visual experience is "out of reach" of explicit descriptions, so to speak.

The application to Searle's relation of "caused by and realized in" is simple. If we know that two events or processes are related causally, it must be because we have an explicit description of them which states what the causal relation is. A consequence of the above argument is therefore that the brain and visual experience cannot be causally related, at least not in the sense which we usually give to that term in science. This is not to deny that there is a connection between the brain, retina, etc. and visual experience; it "merely" says that this relation is not explicitly describable and therefore cannot be called causal.

The contradiction in Searle's replies is that he cannot both refute the counterargument that a simulation of the brain to the smallest level of detail would be conscious and at the same time claim that the mind is caused by the brain. The contradiction emerges when the confusion surrounding the concepts of "caused by" and "realized in" is cleared up.

This conclusion can be rephrased in general terms as the thesis that it is meaningless to say that the mind is implementation-dependent. I call a process a \textit{implementation-dependent} on process
B when any realization of A is necessarily a realization of B as well. The relation between A and B is as between the two levels of computation in figure 8. In figure 8, each lower level process implements the processes above it. Implementation-dependence of process at level \( n \) on a process at level \( m, m < n \), is the phenomenon that each time the process at level \( n \) is realized, the process at level \( m \) is necessarily realized as well.

It is this claim which Searle makes with respect to mental processes and brain processes. My critique of it is not that the claim is false, but that it is meaningless, since a presupposition is not fulfilled which would make Searle's claim have a truth-value: Some mental processes, like visual experience (and explication, as was argued in chapter 2), are not explicitly describable. This view of the matter also applies to the strong AI thesis which Searle attempts to refute. Just as it is meaningless to say that mind depends essentially on the brain for its implementation, so it is meaningless to say that the mind is independent of implementation in the brain. The strong AI thesis is not so much refuted as rendered meaningless.

My view of the Chinese room experiment should by now also be clear: The process by which the man manipulates squiggles is explicitly described, and therefore does not describe a mental process. This view is unaffected by the counterexamples discussed by Searle.

### 3.5. Digression: Implementation-independence and AI methodology

I want to digress briefly in this section from the main theme, the possibility of MI, with a methodological look at the thesis of the implementation-independence of the mind which is central to current AI research. I start with two definitions.

Let me define **natural science** as the study of the state evolution function of systems as they are encountered in nature, and **engineering science** as the study of the way in which a state evolution function thought to be desirable can be implemented. For example, high-energy physics, neurology, and linguistics are natural sciences, because the behavior of the systems they study is described as it is encountered in nature. Bridges, boats, airplanes, and houses are systems studied by engineering science, because their behavior is deemed desirable by people. There are various mixtures of the two types of science in practice. For example, to be able to conduct experiments in high energy physics, a lot of engineering science is used, and in the study of airfoils, new facts are discovered about the natural phenomenon of turbulence. But the distinction between the two research **goals** is clear: In natural science, behavior found in nature is studied, in engineering science, behavior desired by man is studied. The behavior studied by natural science may be very undesirable, and the behavior studied by engineering science may never have occurred before in nature.

Given these differences in research goals, let me place the notion of implementation-independence in perspective. In engineering science, it is immaterial how a state evolution function is implemented, as long as the implementation satisfies certain engineering standards. Important engineering standards are cost-effectiveness (the implementation medium should be cheap), reliability (few breakdowns should occur), robustness (it should function in a wide range of environments) and, above all, usefulness, (the artifact should satisfy a desire of man). All these desiderata are external to the process studied, and they motivate the choice of implementation medium—cheap, strong, reliable, etc. Implementation-independence of the desired behavior is thus a central concept in engineering science because it gives us a freedom of choice of implementation materials. Note that implementation-independence "disembodies" the function to be implemented and therefore requires a clear understanding of the function to be implemented, independently of the implementation medium. It makes no sense to try to implement a machine which "more or less can fly." This ties in with the presupposition of explicit describability argued to be necessary for implementation-independence above. (It may be the case, as often happens in computer science, that people have only a vague idea of what they want. Then prototypes of the desired machine are built and given to the user to play with, so that he can clearly say what he wants. This statement of requirements is then used to draw up a contract which states clearly which functions the desired machine should fulfill.)

In natural science, implementation-independence plays no role except in reasoning by
analogy. When the behavior of electricity is studied, we want to know how electricity is realized in nature, not how we can implement analogous behavior in a system of water pipes. We may try to describe the behavior of one process by borrowing a description known to describe similar behavior (see Harré [1970] for a classification of this type of reasoning). But the research goal rules out implementation-independence as an interesting phenomenon. Note that the behavior of the process whose natural implementation is studied, is often ill-understood. Natural science starts with ill-understood state evolution functions of systems as they are found in nature, not with clearly stated desired behavior.

A science of mind should therefore try to explicate the currently ill-understood state evolution function of mental processes and then study the natural implementation of this in the brain. Assume for the sake of the argument that this research goal is attainable and that all that has been said about explicit describability in this thesis is mistaken. What can we then say about AI research from this methodological point of view?

AI research tries to study ill-understood mental functions independently of the implementation. It thus combines natural and engineering science research methods, but combines them only partly. From natural science it adopts the study of ill-understood functions, and from engineering science it adopts the idea of studying behavior in an artificial implementation. This does not bode well for research results. From a natural science point of view, is the implemented function (which has been explicated some way) an interesting abstraction of the function found in nature? Does the implementation tell us anything about the natural implementation of the function? Debugging a program until it works does not give any clue as to the answer. To some extent, this criticism extends to any study of natural functions by simulation, but the initial vagueness and the distance from the end-result to the simulated behavior seems the largest in AI research. (Cf. McDermott [1976] for some comments upon the scientific value of AI programs.) From an engineering point of view, other questions can be asked. Is the function desirable? Is the implementation cost-effective, reliable, robust? The answer to these questions is negative in the majority of AI programs (with some notable exceptions, like the expert system DENDRAL and the symbolic integration program MACSYMA).

All of this may be written off as unfair and premature criticism of a young discipline. But the analysis of the previous paragraphs shows at least that the question of the justification of the research method in AI is still open, and that it is up to AI researchers to provide one. (Ringle [1983] and Hayes [1984] make some attempt in this direction, but their arguments are far from conclusive and the mixture of natural and engineering science at issue is put into question by Sharkey & Pfeifer [1984]). It also shows that the AI mix of natural and engineering science depends crucially upon the adoption of the idea of implementation-independence from engineering science. As long as this mix of methods leads to results which fall below the standards of natural science as well as engineering science, there is reason to believe that the strong AI thesis of the implementation-independence of the mind is a bad idea to base a research method on, and the longer this situation continues, the more weight must be attached to this reason.

3.6. Dreyfus

Dreyfus [1979] (pp. 100-129) gives a number of arguments why computers cannot be intelligent which are similar in intent but different in structure to mine. On pp. 100-129 of his [1979], Dreyfus gives arguments why he thinks computers cannot execute a process he calls "perspicuous grouping" of data. I summarize these arguments first and then discuss them. Dreyfus assumes that human beings can perspicuously group data, while computers manipulate discrete symbols (ibid., p. 128). This contrast between perspicuous grouping and symbol-manipulation is analyzed into three elements, zeroing in, ambiguity tolerance, and insight.

1. People are able to zero in on a promising aspect of the situation. Chess players do this, without being aware of counting out alternatives. If they did count out alternatives unconsciously, then this must be an enormously efficient computation and it seems weird that shortly before reaching a determinate output, the computation stops being unconscious and switches to a slow, cumbersome, conscious mode (ibid. p106). After zeroing in, a
chess player counts out the (relatively) few alternatives for the move he wants to make. Doing this, he relies on a background of past experience at the "fringe of his consciousness." Computers cannot do this, Dreyfus says, because they must start counting out alternatives in a step-by-step manner already during the process of "zeroing in."

2. Human beings tolerate ambiguity and resolve it by their sense of the situation. Many natural language expressions, for example, are highly ambiguous, but most of the logically possible interpretations do not even come up for consideration because the expressions appear in a context. Computers, by contrast, must manipulate information in context-free bits and can therefore not resolve the ambiguity.

3. The third component of perspicuous grouping is termed alternatively insight, seeing the structure of the problem, restructuring the problem, or separating the essential from the inessential aspects of the situation. This is a major element of human creative thinking and is contrasted with the way a computer must search a space of problem situations. A computer cannot separate the essential from the inessential unless the programmer introduces the distinction in the first place, either in the program itself, or in a planning program, or in a meta-planning program, etc. (ibid. p.118).

Some obvious differences between these arguments and mine are that 1. Dreyfus' argument applies to computers, not to machines in general; and that 2. Dreyfus argues that computers cannot perspicuously group data, not that they cannot explicate.

A consequence of the first difference is that my argument applies not only to the current generation of universal digital computers, but also to non-standard computing devices as the connection machine and the Boltzmann machine.

Connected to this is that Dreyfus's argument does not rest on an explicit description of those properties of a computer which prevent it from executing the processes Dreyfus says it cannot execute (cf. Buchanan [1973], p. 21). The observation that these processes have resisted automation up till now is not sufficient to warrant the conclusion that they cannot be automated. For example from the fact that a computer processes information in a discrete manner by manipulating context-free bits or by searching a space of alternatives it does not follow, without further argument, that it cannot zero in, resolve ambiguity, or (re)structure a problem. To a certain extent, these processes have been executed by computers, but with problems of any size the computations are crushed under a mountain of alternatives which have to be considered explicitly. To show that the combinatorial explosion of alternatives is not a technical problem but one of principle, more argument is needed.

A consequence of the second difference between Dreyfus' argument and mine is that his argument is more controversial and much harder to get right, because the concept of perspicuous grouping is vaguer than that of explication. Despite this extra effort, it would still share a deficiency with my own argument, viz. that there are people who are not able to execute the relevant process. Some people with mental illnesses cannot perspicuously group data and thereby fall outside the domain to which the argument applies. Such people execute processes which probably are even less amenable to explicit description than explication, but because we don't have first-hand experience of those processes, it is even harder to find counterintuitive consequences of the assumption that those processes are explicitly describable.

A final difficulty with Dreyfus' argument is that perspicuous grouping is not defined independently of the contrast with computers. By not separating the different parts of the argument, accepting or rejecting it becomes a kind of all-or-nothing affair in which it is hard to say with which part of the argument one disagrees.

10. If we restrict ourselves to algorithmic processes, the theory of NP-completeness could be used (Garey & Johnson [1979]). But even there it can be shown at most that if an algorithm for a certain problem can be found whose execution time does not grow exponentially with time, then a host of other hard problems are solvable in non-exponential time as well.
A difference of another order between Dreyfus' argument and mine is that he imputes a series of questionable assumptions (called the biological, psychological, epistemological and ontological assumption) to AI researchers, whereas I do not. In reaction to his book, it was promptly denied that AI researchers do in fact entertain those assumptions, or that they need to entertain them, or that it was claimed that it is not a bad thing to entertain them, or that they are not entertained anymore (Buchanan [1973], Williams [1973], Wilks [1976]). Most of the discussion around Dreyfus' arguments has focused on these assumptions or on the supposed maliciousness of his "attack on AI" and this has clouded the important issues mentioned above.

Dreyfus ends his book with a proposal for an alternative method for psychology based upon ideas from phenomenology. I think that it is mistaken to expect results from phenomenological research which can be compared significantly with those of science. In particular, I don't think there exist explicit descriptions relating the results of phenomenological research with those of science. I turn to this and some other consequences of my thesis in the next chapter.
Chapter 4
Some consequences of the thesis

After these skirmishes in various directions, I now return to the main theme of this thesis, explication and explicitness. The conclusion of chapter 2 is that explication is a mental process to which the concept of state is not applicable and which is therefore not explicitly describable. The main reason for the concept of state being not applicable is that anything about the context of explication can be relevant for the outcome of the explication process. In this chapter, this result will be expanded to formulate two related standpoints which I call ontological monism and epistemological dualism. The claims of these two views follow from the two statements below, which in their turn summarize the relevant conclusions from the argument of chapter 2.

1. Explication is a gradual process. There is no sharp division between explicit descriptions and totally inexplicit descriptions. Between these two types of descriptions lies a continuous spectrum of explicitness.

2. The ability to explicate, as well as individual explication processes, are not explicitly describable. Any explicit description of the ability to explicate will fail to capture the open-ended quality that the range of situations which can be explicated is not explicitly demarcated, and any explicit description of an individual explication process will fail to capture potentially relevant aspects of the context of explication (which includes the person who does the explicating).

Statement 2 says in effect that explication is literally inexplicable, and section 1 of this chapter explores some reasons why this should be so. The main source of inspiration for this section is the philosophy of Heidegger. Section 4.2 then formulates the two thesis connected with the graduality and inexplicability of explication. Section 4.3 then applies this to a very practical affair, the use of machines by people.

4.1. Explication revisited

In chapter 1, the outcome of an explication process, an explicit description, was characterized. In order to better understand the statement that there are degrees of explicitness, in this section I turn to the starting point of explication. If we do this, we are in for some grave problems, for suppose that what lies at the start of explication is explicitly describable. Then the explication process would be explicitly describable. This contradicts all that has been argued before, so that, even if my argument is incorrect, consistency demands that I come up with inexplicit descriptions, i.e. descriptions which are incomprehensible, or describe an unrepeatable situation, or are very context-dependent. Any attempt to say something about the start of the explication process therefore goes against a conviction which is expressed by the famous words of Wittgenstein,

\[ Wovon \text{ man nicht sprechen kann, darüber muß man schweigen. Wittgenstein [1973], p. 115. } \]

Against this injunction -against which Wittgenstein was the first to acknowledge that he had sinned- I want set the belief expressed by Heidegger that if we don’t try to describe what precedes explication, then we throw away our capacity as human beings to think (Heidegger [1959a], p. 25). The motto to be followed then, becomes

\[ \text{Beweisen läßt sich in diesem Bereich nichts, aber weisen manches. Heidegger [1957], p. 8. } \]

The nature of the problems we are likely to encounter is more or less characterized when we negate the properties of explicit descriptions, for whatever describes that which explication starts with, it must be the opposite of an explicit description. In other words, it must be an incommunicable and maximally context-sensitive description of an unrepeatable situation. Each of these
three characteristics implies that one person cannot understand such a description when uttered by another. If such a description is understood, it must be understood not to apply to the current situation. But if at least that is understood, something has come across, which is impossible by definition. So it is understood that even inapplicability to the current situation cannot be understood, etcetera. Looking purely at the logic of totally inexplicit descriptions, the process of understanding such descriptions iterates from negation to negation. Totally inexplicit descriptions are inherently instable; the attempt to understand them never ends.

But this unending character of the search for what precedes explication should not stop us from undertaking it. If we do not undertake it, we may forsake, as Heidegger says, what is the essence of human existence, the capacity for thought. We can learn two things from Heidegger's philosophy which are relevant to the phenomenon of explication. First, Heidegger's analysis of human existence reveals some structures in human existence which make human beings the kind of beings who can explicate. Second, the analysis of these structures gives some insight into what it is that lies at the start of the explication process. I discuss these two points in turn.

An entity which can explicate is an entity which can carry out a process of which at least in the initial phase cannot be said to be a sequence of states. There is no finite description of what the effects of possible inputs on the development of the process and its output will have. In other words, we cannot use the language of states, state evolutions, processes, inputs and outputs to describe its peculiarities (the previous sentence, which uses that language, must therefore be taken metaphorically).

One peculiarity of explication is that anything about the environment, including the history of the explicating agent, can be relevant to its outcome. That is what makes it so context-sensitive. Another peculiarity is that to the agent himself it is not exactly clear in advance which aspects are relevant (if it were, explication would already have taken place). The act of explication starts, as it were, from nothing, in the sense that whatever it starts from is not a thing when viewed in the way the subject sees it at the start of explication. This is true for someone brooding over the next sentence in a written argument, a photographer looking for the right composition, right moment and right dynamics of his photo, a mathematician trying to find a proof, a programmer trying to locate a bug in a program, or two friends of which each explains the other's moods to the other. In some of these cases, explication goes so far as to lead to an explicit description, and in some of those cases, the clues which lead to the explicit description may be described as definite things separable from their environment. But as such they did not lead to the explication. On the contrary, it is explication which made them appear as such. For this reason, explications of the clues which lead to an explicit description always seem to leave a significant but indeterminate residue which also was relevant.

We can put this in partly in the language of systems theory (as was done in the previous paragraph), partly in the language of Heidegger (as will be done in the next). What has just been said is that explication literally starts from a significant nothingness. Now, a thing is something which is conceptually separable from its environment, can be in a state and can go through a process. In short, it is what I have called a system from the first sentence of section 1.1 up till now. Things may be material, but mental processes or organizations of people, in as far as they can be described as systems, are things as well. To say that explication starts with

1. This phenomenon is used in Buddhist texts in a kind of dialectical progression of statements and their contradictions in order to evoke understanding in the reader (e.g. the Heart Sutra, Conze [1958]).
2. This phenomenon is central to Buddhism as well. See Gunaratne [1986].
3. The limit of this infinite series is a very special description, a silent expression of unutterable truth. It may be, however, that the series does not converge to one finite but has as many limits as there are occasions to be silent.

As I understand Buddha's meaning there is no formulation of truth called Consummation of Incomparable Enlightenment. Moreover, the Thatagata has no formulated teaching to enunciate. Wherefore? Because the Thatagata has said that truth is uncontainable and inexpressible. It neither is nor is it not. (Diamond Sutra, Price & Wong Mou-Lam [1969]).
nothing is to say that it does not start from a system. To say that it starts with significant nothingness is to say that this, which is not a system, can be highly relevant.

Changing now to the language of Heidegger’s phenomenology, to say that what explication starts from is not a thing is to say that it lies beyond the horizon of the current situation. A being which can explicate must be open to what is beyond the current horizon; explicating it, it is drawn within the horizon and this at the same time opens new horizons. Heidegger calls a being which is not simply sensitive to well-described inputs but to the significant nothingness beyond the horizon of the current situation a Dasein. A Dasein is a being which is there, which is shorthand for “to be in a world and be open to its horizon.” Using a language which suggests some of the darkness which precedes explanation, Heidegger describes it thus:

Nur auf dem Grunde der ursprünglichen Offenbarkeit des Nichts kann das Dasein des Menschen auf Seiendes zugehen und eingehen. Sofern aber das Dasein seinem Wesen nach zu Seiendem, das es nicht ist und das es selbst ist, sich verhält, kommt es als solches Dasein je schon aus dem offenbaren Nichts her.

Da-sein heißt: Hineingehaltenheit in das Nichts. (Heidegger [1929], p. 114)

There is an original openness to the situation and its horizon which makes it possible that man can be directed at things and that he can be absorbed by his world. This openness precedes in the ontological order of things that which it opens up, the everyday world of average-sized dry goods. Dasein is open to Nichts or nothingness.

The image of a horizon may suggest that the source of what comes upon us in the act of explication lies far away. At several places, Heidegger emphasises that the opposite is true: It is closer to us than the explicitly described things we put before us.

It is so close because it precedes our explicit descriptions and that is why it is so difficult to think about. What Heidegger calls thinking is an effort to experience this nothingness in its closeness. The reason to attempt to think the unthinkable is that the unthinkable makes it possible that we dwell with the things in the world, the things we describe by more or less explicit descriptions. If we confine ourselves to the explicitly thinkable, Heidegger says, we ignore the essence of what makes us human beings, i.e. beings who are there.

A beginning of an answer to the first question posed above, what kind of being it is that can explicate, has been made by saying that it is a Dasein which is open to the significant nothingness beyond the horizon of the current situation; this horizon lies closer to us than whatever is described by explicit descriptions. To some people, this is meaningless language. This is as it should be, for it would be self-contradictory to describe what precedes explication in explicit terms. Being inexplicit, the descriptions we come up with have at least one of the properties of not being communicable to all people, or not being context-independent, or not describing repeatable processes. They may therefore strike some people as meaningless.

The answer could be worked out much more by following the structures of Dasein described by Heidegger [1927]. This will not be done here, since it will exceed by far the bounds put on this essay. But enough has been said to show that an opening exists which connects the world of science and systems theory with the world of metaphysics and even mysticism. And it is man which is the bridge between the two worlds.

4. In another context Heidegger calls it emptiness (Heidegger [1959], p. 108), thus creating a link with Buddhism.
Something has been said already about the second question, what it is which lies at the start of the explication process. It has been described as significant nothingness; Heidegger uses also the term world for it, which is not the totality of systems which are discernable by man, but that which precedes the systems thus discerned. Man dwells in a world as a web of meanings which point over the current horizon and as a set of familiar ways of acting. Man has to be familiar with this world in order to describe the things he meets in everyday life (Duintjer [1966], p. 188). Because of this close connectedness between man and world, Heidegger calls Dasein In-der-Welt-sein. This does not mean that the world is part of man and is but the projection of his ideas, nor that man is in the world as a pen is in a box. We cannot take man out of his world as we can take a pen out of a box, and we cannot destroy the world as we can stop a movie by stopping the projector. What it does mean is that man is ever "outside himself," open to the world and acting in it, meeting it and feeling its resistance. On the other hand, the dwelling and acting in the world can only take place "inside the world," in a field provided to him by the world (Duintjer, ibid.). This interplay between outside and inside is epitomized by calling Dasein a Being-in-the-world.

To sum up what has been said so far, explication produces descriptions of systems and takes its clues from what lies beyond the horizon of the current situation. What lies beyond the horizon is the world we live in. This is so close to us that it precedes explicit description and must be described by phrases as "significant nothingness." This account will be extended by one more point.

By definition, the world in the sense Heidegger takes it is not explicitly describable. But the world is there before and after we produce an explicit description of one of its aspects. Strictly spoken, it is not an "it" at all in the sense that it has a constant identity which endures through time and change. The world changes with our explications to form new contexts and new horizons, but this, too, is a metaphorical way of using "the" and "changes," for I am not using these words in the meanings they have in "the system changes state." We can put the point in a different way by saying that there is always a context to our descriptions from which more can be explicated. To any formal language, there is always an informal metalevel. This property is bound to the elusiveness of the concepts of world and nothingness. When I explicate the concept of a world, I am left with nothing, but it is still there in the background.

This concludes our detour through Heidegger's philosophy. I will now use this analysis to define two standpoints which can serve as a context for the arguments about explication in the previous two chapters, ontological monism and epistemological dualism.

4.2. Ontological monism and epistemological dualism

In the introduction to this chapter it was noted that explication is a gradual process. In section 4.1, explication has been described as rising out of the world, which in its turn is described as significant nothingness. In section 1.1, systems, which are described by the outcome of an explication process, were described as any part of reality which we are interested in and which can be in a state and go through state changes. Putting this together, there is thus a gradual ontological transition between the nothingness we are directed at at the start of explication and the systems we are describing after explication. The transition is called ontological, for it concerns the reality we find ourselves in, not the ideas we have about reality. We have no control over the clues which inspire us during explication, so the world with which explication starts is not a construct of our mind. It is not part of our imagination. And the systems we describe are not figments of our imagination either, since they can be observed and experimented with.

The gradual transition from the initial nothingness to the explicitly described systems is thus a gradation in ways of being. There is no clear line to draw between the nothingness and systems. On the contrary, there is a surprising agreement between the two. Systems are described functionally, by giving a state evolution function which describes the effect of inputs on the state and output of the system. Now, it is a peculiarity of state evolution functions that they can, in principle, be applied to systems made of different "substance," to use the old-fashioned term, or "implementation medium," to use the term common in engineering science. A well-known
example, mentioned earlier, is the state evolution function for the behavior of electricity in a conductor, which is applicable to the behavior of water through pipes as well. The reason why this is so in particular cases or in general does not interest us here. Important for the present purpose is that, in principle, any state evolution function may be applicable to any part of nature. If a state evolution function is applied to systems of one type of "substance" only, it is because we have not run against examples of a type of system to which it could be applied as well. 5

Dropping the language of substances, state evolution functions describe a system at one level of aggregation, and there may be different types of subsystems which can be aggregated to form a system which behaves that way. The differences among the subsystems in some way is balanced out by the way they are aggregated, so that aggregate systems composed of very different subsystems behave according to the same function. Water molecules and electrons are very different types of systems, but there are aggregates of both which behave the same way.

What this amounts to is that a state evolution function characterizes a system in a disembodied way, by not looking at the lower levels of aggregation. What is more, the concept of substance has no place in the description of systems. When we move down in the level of aggregation, we encounter subsystems for which the same principle holds: They are described in a disembodied way, and outside this description there is nothing relevant to say about them. 6 And there is no principled lower bound to the level of aggregation (if the money needed to do experiments in high-energy physics is proportional to the energy required to do an experiment, then there is a pragmatical bottom line). But even if there were, the state evolution function of the smallest particle would still, in principle, be applicable to other systems -any other system, from international finance to problem solving to the solar system. Systems are peculiarly empty. The concept of substance is old-fashioned and has no place in modern science.

We come to the conclusion that systems are empty, while we saw already that the world, as that which precedes explication, is empty as well. The ontological monism this leads to is that reality, through which no explicit dividing line can be drawn so as to divide explicitly described systems from that from which explication starts, is empty. 7

But this ontological monism is bound to an epistemological dualism. The second statement in the introduction to this chapter stated the conclusion of chapter 2, that explication is not explicitly describable. And section 4.1 ended with the remark that to every explicit description, there is always a context which can be made explicit, and when we try to explicate that context, there is more to explicated in the context of that description, etc. With the introduction of an explicit description in the situation, the horizon shifts to open new possibilities. There is never an end to explication. Though reality is one, we can only describe part of it. What is unexplicable is not unreal: We just cannot describe it explicitly. 8 These conclusions bear well with each other:

5. The principle of implementation-independence, central to engineering method, is thus closely connected to the concept of a state evolution function. This does not contradict my earlier statement that natural science does not follow the principle of implementation-independence. Natural science studies behavior as it is implemented in nature, and this goal forbids using the principle of implementation-independence in natural science research. But as pointed out earlier, that principle does have another use in natural science, as part of the justification of analogical reasoning.

6. The description of nature as a collection of systems came into its own with Newton, who found a way to write down the state evolution function of continuous systems (as differential equations) and at the same time consolidated the experimental method. What was lacking in the constructivity in the mathematics he used, was thus made up for by the repeatable constructability of his experiments independent of the peculiarities of the person doing them. Dijksterhuis [1961] describes the transition from medieval to modern science as a transition from thinking in terms of substances to thinking in terms of behavior patterns and functions. I am not saying, of course, that Newton invented the language of systems theory (or discovered the experimental method all by himself), but I do think his achievement can be validly explicated as such. Cf. chapter 1 note 2.

7. This monism is inspired by, but not equal to, the Buddhist philosophy of emptiness: The highest reality is empty, but the realization that this is so is also the realization that the common cycle of life and death is empty.

8. Compare this with Thomas Nagel's standpoint that various forms of reductionism are "motivated by an
What can be explicitly described are systems, not what precedes explicit description. When we attempt to describe what precedes it, we embark on a never-ending explication.

The descriptions we talk about are expressions of knowledge. Science builds up knowledge expressed in explicit descriptions of systems, and in our everyday acting in the world we have unexpressed knowledge of the world which precedes explication. Apparently, there is a duality in knowledge, for explicit knowledge cannot have the unexplicated world as an object. When it attempts to describe it, the world retreats and we are left with just another system. On the other hand, working with explicit descriptions and the systems they describe creates an unexplicated context which is the source of inspiration of new discoveries in the search for explicit descriptions of nature. This creates an inequality between explicit knowledge and knowledge which precedes explication. It is this inequality which I call epistemological dualism.  

Epistemological dualism is bound to ontological monism, for it is because of the dualism that we could not draw a clear and explicit line which divides reality into two. And it is because of our unexplicated familiarity with the world and our sensitivity for its horizon that we know that there are two types of knowledge.

It is interesting to see what this standpoint does to the traditional problems of mind/body duality. Haugeland [1981] neatly sums up these problems:

(i) the metaphysical problem of mind interacting with matter;
(ii) the theoretical problem of explaining the relevance of meanings, without appealing to a question-begging homunculus; and
(iii) the methodological issue over the empirical testability (and hence, respectability) of "mentalistc" explanations. (p. 2)

Problem 1 is no problem at all in as far as science studies just the behavior and state evolution of systems. The nature of the subsystems is immaterial to this study. Concepts like mind and body have no place in it.

Problem 2 can be dealt with in the same manner: Explicitly described systems do not "have" meanings in the relevant sense and there is no state evolution function where any part of the state evolution must be explained by an appeal to meanings. If it had to be described in that way, it would not be a state evolution function. Like one network chatter on the Mod.ai network (an international computer network devoted to discussions on AI) recently said, no computer program has ever been debugged by using the concept of consciousness.

It should be no surprise that problem 3 vanishes into thin air in the same way. State evolution functions describe the relation between input, output, and state evolution. By definition, systems behave observably (albeit sometimes in a very indirect way) or else they are not systems.

epistemological criterion of reality—that only what can be understood in a certain way exists.” (Nagel [1986], p. 15) Elsewhere in the same book he calls it "scientism" (p. 9) and "physicalism" (p. 26). I would explicate the epistemological criterion for reality as “only what can be explicitly described exists.”

9. The duality in knowledge is inspired by, but not equal to, the Buddhist philosophy of two truths, one absolute, concerning the emptiness of things, the other relative, concerning the common cycle of daily activities. The philosophy of two truths is itself an example of relative truth.
10. Harre [1970] also argues for a kind of epistemological dualism when he says that neuropsychology cannot classify brain states without the help of first-person reports about mental states of the subject (pp. 209-227). I or my close friends can report about my mental states (p. 222) and thus provide information about brain states which would not have been obtainable from the study of brain states alone. As a consequence, the subject is recruited to the scientific team (p. 224). The advance of explicit knowledge of the brain requires the use of implicit knowledge of the subject and, as I argue in Wieringa [1984], of the experimenter as well. This is a peculiarity of the study of mental functions.

11. Having cleared the goal free of defenders, Haugeland then scores as follows: "The computational idea can be seen as slicing through all three dilemmas at a stroke; and this is what gives it. I think, the bulk of its tremendous gut-level appeal." Everybody clap your hands.
These answers say nothing more than that science should study mental behavior in the same way as it studies other behavior. But unlike the behavior of other systems, to human behavior belongs explication and this creates the research goal to describe explication explicitly. But explication is not explicitly describable. Hence, concepts like meaning creep in, which have a place in our everyday familiarity with the world, but which are not explicit and do not have a place in explicit descriptions. There is an intrusion of implicit, unexplicated knowledge into explicitly expressed knowledge which no cognitive scientist has yet been able to eliminate.

According to epistemological dualism, this cannot be avoided. There is not one explicit description linking a concept like explication, which derives its meaning from the place it has in the world of unexplicated behavior of human beings, to concepts which derive their meaning solely from their role in the description of observable behavior and state evolution. Rather than attempt the impossible and try to eliminate ineliminable terms, it as well to accept this situation and investigate why these terms are ineliminable and in what way they occur in the descriptions.

Fodor [1983] made an interesting observation which also leads to the conclusion that certain mental processes cannot be described in the language of science. He starts from the observation that higher mental processes are global, by which he means that anything can be relevant to them (cf. the main reason why explication is not explicitly describable. Fodor uses a different argument, a supposed analogy between higher mental processes and scientific confirmation. Anything can be relevant to the confirmation of a scientific hypothesis, Fodor says, so by analogy anything can be relevant to higher mental processes. (Fodor op. cit. p. 105). From this he argues for what he calls "Fodor's First Law of the Nonexistence of Cognitive Science:"

The more global a cognitive process is, the less anybody understands it. Very global processes, like analogical reasoning, aren't understood at all. (p. 107).

According to Fodor, higher mental processes cannot be neatly encapsulated and studied in isolation, but peripheral processes can, because these are reasonably modular. The greatest advances in neuroscience have been made in the study of the visual system, auditory system, motor system etc. because these can be studied in isolation. What has been achieved is the "the ghost has been chased further back into the machine, but it has not been exorcised." (p. 127) According to Fodor, this is not accidentaly but necessarily so.

This is in agreement with my conclusion that mental processes cannot be described in the language of state evolution functions. The distinction between systems describable by state evolution functions and the rest of the world being vague and gradual, one would expect that advances can be made in an area where most processes are still of a kind which has proven to be amenable to description by state evolution, i.e. physiological processes such as the peripheral processes of the neurosystem. I would add, though, that the goal of exorcising the ghost from the machine is not realistic. It would be more fruitful to study the way how it is embodied in the machine -abandoning, of course, the goal of finding an explicit description of how it is embodied.

4.3. Dualism between man and machine

The dualism between explicit descriptions and the inexplicit world which explication starts from can be viewed from another angle. Explicitly described systems are machines, and industrial man is in the habit of making machines. In this section, I will draw some conclusions for engineering in general and computer science in particular. The conclusions I will draw are summed up in the following quotation from Winograd [1981].

A person writing a program (or contributing it its "knowledge base") does so within a background of assumptions about how the program will be used and who will be interpreting its

12. See chapter 2. note 6.
responses. Part of this can be made explicit in documentation, but part is an implicit background of what can be "normally understood." Except for systems operating within highly constrained domains, there inevitably comes a time when the system "breaks down" because it is being used in a way that does not fit the assumptions.

There is always a limit set by what has been made explicit, and always the potential of breakdowns that call for moving beyond this limit. (Winograd [1981] pp. 258-259)

To which I am inclined to add, "etcetera." If the cause of the breakdown has been located and made explicit, a procedure for handling it can be added to the explicit description. The remark made by Winograd then applies to this new description, etc. In an important sense, the background is infinite while our explicit descriptions are finite. The "implicit background of what can be 'normally understood'" is Heidegger's world. 13

Winograd & Flores [1986] point out an aspect of Heidegger's early philosophy which is relevant to the mechanization of intelligence. Tools-useful artefacts- are at hand (Zuhanden) without the user directing his or her attention at them. Attention of the user is directed at a goal, which lies in a different domain altogether from the domain which the manufacturer of the tool lives in. For the user, tools are "just there," subject to an implicit web of meanings, and they only appear explicitly in the domain of the user when they malfunction (Winograd & Flores [1986] p. 36, Heidegger [1927] p. 74, 361). The implicit world we live in is only explicated when needed, i.e. when the web of what is normal and expected is ruptured.

This has a number of consequences. First, computer programmers rarely realize that the users of their programs are not interested in the sources of errors in a program. When a program is used, the user is directed at one domain, but when the program breaks down, a completely different domain emerges. These two domains may be so unrelated as political history and Pascal programming, if the user wrote a paper on political history and the word processor is written in Pascal. Most people find such an abrupt and unwanted change of domain unpleasant. To a programmer using the word processor for a programming-related task, on the other hand, the change is not nearly as abrupt. The implicit web of meanings he dwelled in at the moment of breakdown concerned programming already. He may very well be interested in the source of the breakdown and talk it over with the writer of the word processing program. Programmers are not exposed to the abrupt and unpleasant change of domain which user outside the domain of computer science are exposed to, and this can create a source for misunderstanding and frustrated communication.

A second consequence is that a program, which is an explicitly described process which functions on unspoken assumptions about its use, is bound to break down when these assumptions are violated. Given this fact, it is better to make the user aware of these limitations than to create the illusion that these limitations are not there, such as all too often is the habit in presentations of AI programs (see McDermott [1976]).

A third consequence follows from the second but is more general, because not limited to computers. Because machines are explicitly described, their use is explicitly described (or at least, the range of uses within which the manufacturer can guarantee proper functioning us explicitly described). In general, the user must be aware of the limits of possible use. But the user is always (possibly via a chain of intermediate machines) a human being, so whenever in an organization a function is mechanized, it is man who must adapt himself to the behavior of the machine. This need not be dramatic at all. For example, it may be an adaption which is required by a needed reorganization anyway, or the machine may relieve the human being from a lot of work, or it may be designed in a very user-friendly manner. But even in such idyllic situations the limits of the machine are determined by its explicit description and it is man who must adjust himself, thus introducing some rigidity-context-independence, repeatability- in the behavior of the users of the machines. This is independent of the degree of user-friendlyness

13. Which, by implication, is infinite.
built into the machine. It may be experienced as unpleasant by the users, regardless the user-friendliness of the machine.

A final point to be made about the use of machines does not follow from the properties of breakdown but from their context-independence (which is also the source of rigid breakdown patterns). The output of a machine is defined in terms of inputs and state sequences but is interpreted by the user in a context of purposes, unspoken assumptions, and interests. To put it in the language of section 4.1, for every explicit description there is an unexplicated metalevel. To assume that a machine shares this context or metalevel is to misunderstand the nature of machines. Winograd & Flores [1986] explain this elaborately in the case of computers. There is an endless number of examples of this in the realm of engineering within as well as outside computer science. Kent [1978] gives many examples in the area of computer science. The examples all illustrate the simple point that

When a data file exists to serve just one application, there is in effect just one context, and users implicitly understand that context; they automatically resolve ambiguities by interpreting words as appropriate for that context. But when files get integrated into a data base serving multiple applications, that ambiguity-resolving mechanism is lost. The assumptions appropriate to the context of one application may not fit the contexts of other applications. (Kent [1978], p. 3)

It is as simple as that, but the consequences are momentous. In the worst case, the integration of two files used in different contexts is useless to all parties concerned.

Outside the realm of computers, a neat example of the importance of the shifts in the context created and perceived by human beings is given by Ottevangers [1986]: In a certain area, a waterpurification installation is built with a capacity sufficient for current and future water needs. To finance the installation, a small tax is levied. Because of this tax, local companies find it profitable to purify their water themselves, so that they don’t have to pay the levy. The result is an installation which is much too large for the remaining community of users and which cannot be financed out of the taxes which this community raises. What has happened is that human users perceived this machine in a context, explicated a profitable alternative and followed that up. Out of nothing, a way is found to circumvent the use of a machine altogether. A bit less extreme is the creativity human user always seem to be able to muster in finding unexpected ways to use machines. E.g., access registration, time clocks and other little machines to get a handle on what is really happening in an organization are used in such a way that the employee constructs a favorable profile of himself in the database into which data from all these machines is poured. All of these examples can be understood to be ineliminable by keeping the gap between context-independence of mechanical processes and the context-sensitivity of human intelligence in mind (this is relevant if we consider, say, the attempts of governments to eliminate tax-frauds, a part of the informal section of society, by automated information processing, which is an application of explicit descriptions. Instead of eliminating tax-frauds we may end up with increased control over a large part of society which followed the rules anyway.)

I cannot resist closing this chapter by quoting from the former president of the International Federation of Information Processing, Zemanek.

Today very frequently the gap between formal and informal universes is overlooked, neglected, ignored. Formal thinking is applied to relations which would require informal understanding and formal structures are very sloppily described in an inadequate and misleading natural language . We deal with the individual and with society sometimes as if they were constructs, but at the same time we let technology grow into the natural environment without planning and control. (Zemanek [1972], pp 138, 136)

The idea that machine intelligence is possible is at the center of the confusion deplored by Zemanek.
5.1. Summary

1. Explicit descriptions are *communicable* to a group of people of arbitrary size. Any one member of this group can, barring practical limitations of time, space, money, required background knowledge, etc., show his understanding of the description by applying it to the described state or process. Understandability by an arbitrary group of people presupposes that the process or state description is *repeatable*, i.e. whenever the initial conditions occur, the process develops as specified by the process description and whenever the conditions in the state description occur, the state is present. This in turn implies that the description is *context-independent*, i.e. all relevant elements of the context are stated in the description and whatever is not stated is not relevant.

2. A machine is a system with a state transition or evolution function. The behavior, processes and states of a machine are explicitly described. This includes, of course, Turing machines, for they execute communicable, repeatable and context-independent descriptions by definition. But it includes also any natural system of which we know the state transition or state evolution function, like the solar system or a falling rock. And it includes any artificial system which is constructed according to an explicit description, like radios, cars and telephones. There are also natural and artificial systems of which we currently do not know an explicit description at an interesting level of abstraction (or detail), such as the weather, two people talking to each other, the !Kung bushmen fire dance, and quarks.

3. Explication is a process which produces an explicit description of a state or process without receiving an equivalent description as input. The *ability* to explicate is the ability to produce an explicit description of a range of states and/or processes which is not explicitly described in advance of explication. Individual cases of explication can succeed to a greater or lesser degree, and human beings differ in their capability to explicate well. But the *ability* to explicate a variety of states and processes which is not circumscribed explicitly in advance is part of human intelligence. In contrast, by definition the range of situations a machine can produce explicit descriptions of is explicitly demarcated in advance of the production of explicit descriptions by the machine.

4. Moreover, particular instances of explication by human intelligence cannot be explicitly described either, because 1. the initial phase of explication is one, if not the, most incommunicable process occurring in human intelligence; and because 2. anything in the environment can be relevant for the explication, it cannot be described in a finite, context-independent way; and because 3. anything in the history of the persons who do the explicating can be relevant to the outcome of the process, repeated occurrences cannot in general be captured by a finite description.

5. In a discussion of related viewpoints, Turing’s arguments for the possibility of MI turn out to be particularly weak, for almost exclusively in the defensive (of the type “How could it be otherwise”). His only positive argument, the possibility of a learning machine, begs the question. His argument would gain in plausibility if some reason would be given to believe that the resulting artifact would be a machine, i.e. explicitly describable, but he does not give us any.

5.2. Arguments revolving about “semantic information processing” conflate “representational content” with “interpretation” in the standard predicate logic sense and lead to the assumption of a homunculus in human mental processes. The need for a homunculus is not obviated by appeal to the automatic nature of some computational processes, for what is at issue is whether the relevant computational processes, those
simulating higher mental functions, exist at all, i.e. whether the relevant mental processes are explicitly describable. Cognitive science and AI, like Turing, issue a promissory note that they are, but have offered nothing in support of this claim. My argument that the note is void therefore stands unaffected.

5.3. Searle formulates a crucial assumption of AI research, the implementation-independence of mental processes. He formulates an interesting counterclaim, that of implementation-dependence of mental processes on the brain. A presupposition which must be satisfied for both claims to have a truth-value is that mental processes and brain processes are explicitly describable. Since I argue that at least one mental process is not explicitly describable, it follows that both claims are meaningless for at least that process.

5.4. My argument can be construed as a support for Jefferson’s claim that machines cannot think and as a cleaned-up version of one of the arguments which appear in Dreyfus [1979]. Whereas Dreyfus concentrates in that particular argument on perspicuous grouping, I concentrate on explication and therefore am able to base my argument on a rather uncontroversial definition of machines. Dreyfus’ argument, by contrast, is limited to computers.

6. Explication is inspired by what Heidegger calls alternatively "world" or "nothingness," the unexplicated reality which we pre-reflectively dwell in. Explicit descriptions describe systems, and the concept of system can be viewed as an explication of the very general concept of entity or thing. The world is not the totality of things, but precedes things, and is therefore aptly called no-thing. On the other hand, systems are described in an essentially disembodied way by state evolution functions, which concentrate on behavior and states, not on what subsystems a given system is composed of. Ontological monism says that 1. there is no sharp dividing line between the world from which explication starts and the collection of systems described by the result of explication, and that 2. the reality at both ends of this scale is empty.

7. Ontological monism goes with epistemological dualism, for the knowledge which we, as executors of the explication process, have of what stands at both ends of the gradual scale of explication, is itself not explicitly describable. Explicit descriptions are limited to the outcome of the explication process. Explicitly expressed knowledge has its limits just as implicit knowledge has, and the two are clearly distinguishable. Because the world is not a thing, it is not explicitly describable, and moreover the context of every explicit description is an unexplicated world. To every explicit description there is always an unexplicated metalevel which contains pointers beyond the horizon of the current situation.

8. In the study of inanimate nature, all relevant domain knowledge has always been expressible in explicit description, but in the study of higher mental processes, concepts drawn from our pre-scientific experience as intelligent beings invariably crop up. The argument of this paper implies that this is necessarily so, for some higher mental processes are not explicitly describable; knowledge of them will always contain a literally inexplicable component.

9. In the construction of useful artifacts -tools-, the possible uses within which proper functioning can be guaranteed are sharply circumscribed. Rather than suggest that there are no sharp boundaries, as is the habit in AI research, the user should be made aware of these limits. The introduction of machines in an organization inevitably introduces some rigidity in the way human beings must deal with the machine, and the desirability of this rigidity must be weighed against the considerations which motivated the introduction of the machine. Moreover, since the unexplicated context in which the machine functions is changed by the introduction of the machine, and because this context is accessible to man but not to the machine, automation can have quite unexpected and undesirable consequences which all have to do with the creative way in which man perceives the introduced machine in a context.
5.2. Conclusion

Patrick Winston, head of the Walhalla of AI technology, the MIT AI lab, says in his popular textbook on AI that

As soon as a process is dissected, studied, grasped, the intelligence invariably seems to vanish ... Vintage performance becomes *vin ordinaire* once details are exposed and limitations are seen. One must recognize this natural tendency or it will lead to a poor attitude. (Winston [1977], p. 254)

The foregoing chapters give support for this claim and investigate some aspects of the poor attitude which results from ignoring it. The arguments to support the claim are not a *proof*, and it is essential for them that they are not. If they were, they could be automated. In this respect they are not unlike the mediæval existence proofs of God, which were designed not so much to convince unbelievers as to explicate what believers knew. The faith supported by these arguments is, precisely, that what it is for a human being to exist in a world can never be exhaustively verbalized. Man's capacity for language raises him above other animals, but the new medium immediately makes him aware of its finiteness, i.e. of the distance between verbalizations and what is verbalized. What is worse, the finiteness cannot be exhaustively verbalized. That does not mean that we should stop talking. On the contrary, it is that finiteness which keeps us talking.

I have tried to argue my thesis across the boundaries of the two cultures of computer science and philosophy (and, within philosophy, across the boundaries of the block-headed hardliners and the romantic nitwits). The Zen question, asked while the master raises his stick, "What was your face before you were born? Answer quickly!" can be translated into the post-industrial question "What is left of this thesis after the solar system has burnt up? Motivate your answer." asked while there is only 5 minutes left in the exam. At least, this thesis has contributed to the heat which, according to the second law of thermodynamics, will be all that is left over.

---

1. Cf. St. Anselm's remark in the Proslogion, written about 1077-1078: "Consequently. I have written the little work that follows, dealing with this and one or two other matters, in the role of one who strives to raise his mind to the contemplation of God and seeks to understand what he believes." Anselm [1957], p. 391.
References

Anderson A.R. (ed.) [1964]

Anselm, St. [1957]

Buchanan B.G. [1973]

Codd E.F. [1979]

Conze E. (transl.) [1958]

Davis M. [1958]

Dennett D.C. [1971]

Dennett D.C. [1978]

Dennett D.C. [1981a]
Brainstorms, Harvester, 1981.

Dennett D.C. [1981b]

Dreyfus H.L. [1979]

Duintjer O.D. [1966]

Dijksterhuis E.J. [1961]

Feynman R.P., R.B. Leighton, M. Sands [1963]
The Feynman Lectures on Physics, Addison-Wesley, 1963.

Fodor, J.A. [1968]  

Fodor J.A. [1983]  

Garey M.R. & D.S. Johnson [1979]  

Gunaratne R.D. [1986]  

Gunderson K. [1985]  
Mentality and Machines, 2nd. edition, Un. of Minnesota Press, 1985

Haugeland J. (ed.) [1981]  

Harré R. [1970]  

Hayes P.J. [1984]  

Heidegger [1927]  
Sein und Zeit, 1927, 1979, Max Niemeyer Verlag.

Heidegger M. [1929a]  
"Was ist Metaphysik?" in Wegmarken, Vittorio Klostermann, 1967, pp. 103-121.

Heidegger M. [1957]  
Identität und Differenz, Neske, 1957.

Heidegger M. [1959a]  

Heidegger M. [1959b]  

Hofstadter D.R. [1979]

Hollingdale H. & G.C. Tootill [1975]

Hopcroft J.E. & J.D. Ullman [1969]
*Formal Languages and their Relation to Automata*, Addison-Wesley, 1969.

Jefferson G. [1949]

Katz, R. [1982]

Kent W. [1978]

LaChat M.R. [1986]

McDermott [1976]

Minsky M. [1967]

Minsky M. [1982]

Miura I., R. Fuller Sasaki [1965]

Myhill J. [1952]

Nagel T. [1986]

Price A.F., Wong Mou-Lam (transl.) [1969]
Ringle M. [1983]

Robinson G. [1972]

Rogers H. [1967]

Ryle G. [1949]
The Concept of Mind, Hutchinson, 1949.

Searle J.R. [1980a]

Searle J.R. [1981b]

Searle J.R. [1983]

Searle J.R. [1984]

Sharkey N.E., R. Pfeifer [1984]

Shelly M. [1818]
Frankenstein, or, the Modern Prometheus, Illustrated by Barry Moser and with an Afterword by Joyce Carol Oates, University of Berkely Press, 1984.

Suzuki D.T. [1960]

Tanenbaum A.S. [1976]

Turing A.M. [1950]

Wieringa R.. [1984]
"Fenomenologische Opmerkingen over Metingen van Hersenmechanismes van Visuele...

Wieringa R.J. [1986]

Wilks Y. [1976]

Williams B. [1973]

Winograd T. [1981]

Winograd T., F. Flores [1986]

Winston P. [1977]

Wittgenstein L. [1973]

Yazdani M., A. Narayanan (eds.) [1984]


Zemanek H. [1972]