Thinking and Computing: Computers as Special Kinds of Signs

James Fetzer

The Commens Encyclopedia

The Digital Encyclopedia of Peirce Studies

New Edition

Edited by Mats Bergman and João Queiroz

URL	http://www.commens.org/encyclopedia/article/fetzer-james-thinking-and-computing-computers-special-kinds-signs
Retrieved	05.02.2025
ISSN	2342-4257
License	Creative Commons Attribution-NonCommercial-ShareAlike

Abstract:

Cognitive science has been dominated by the computational conception that cogniton is computation across representations. To the extent to which cognition is supposed to be a purposive, meaningful, algorithmic, problem-solving activity, however, computers appear to be incapable of cognition. They are devices that can facilitate computations on the basis of semantic grounding relations as special kinds of signs. Even their algorithmic, problem-solving character arises from ther interpretation by human users. Strictly speaking, computers as such–apart from human users–are not only incapable of cognition but even incapable of computation, properly construed. If we want to understand the nature of thought, we are going to have to study thinking, not computing, because they are not the same thing.

Keywords: Cognition, Cognitive Science, Computers, Kinds of Minds, Kinds of Signs, Algorithms, Thinking and Computing

Perhaps no conception has ever dominated a discipline as the notion of a computer has dominated the science of cognition. Predicated on the analogy that the mind stands to the brain as software to hardware, the computational conception has assumed the status of a paradigm, where normal cognitive scientists devote themselves to its defense and elaboration, even when confronted with an increasing accumulation of anomalies in the form of problems and objections with which it cannot readily be reconciled. Publications within this paradigm abound, including such familiar works as *Cognitive Science: An Introduction*, co-authored by Neil Stillings and other cognitive scientists (Stillings, 1987), and Barbara von Eckhardt’s *What is Cognitive Science?* (von Eckhardt, 1993).

Books like these from The MIT Press—the leading publisher of work within this domain in the world today—are representative of this approach in different but related ways. The Stillings volume qualifies as an unrelenting endeavor to integrate virtually every cognitive activity—from perception to memory to reasoning and beyond—within the computational paradigm. The von Eckhart book is even more striking as a bold attempt to preempt the very possiblity of non-computational cognitive science by means of stipulation, where non-computational phenomena cannot qualify as “cognitive” as a matter of definition. This stance has not muted its enthusiastic reception: it has even been described as “the best philosophical discussion of cognitive science to date” (Thagard, 1995).

The paradigm has exerted such immense appeal to so many for so long that its limitations have become difficult to discern. My purpose here, therefore, is to subject the computational conception to a severe test of (what I take to be) its most elementary ingredient, the supposition that *cognition is computation across representations*. Since computers are or at least appear to be capable of computation across representations, it follows from this supposition that computers are or at least appear to be capable of cognition. To the extent to which computation across representations is supposed to be a purposive, meaningful, algorithmic problem-solving activity, however, computers appear to be incapable of cognition. They are appropriately envisioned as devices that can faciliate computation on the basis of semantic grounding relations as special kinds of signs.

What is Cognitive Science?

As an exercise in methodology, von Eckhard's approach clearly deserves to be considered. She not only adopts the substantive assumptions that cognition is *computational* and that it is also *representational* but explicitly embraces the strategy of excluding from consideration non-normal, non-adult, non-humans (von Eckhardt, 1993, p.6-9). Her approach thus not only precludes the study of the young and the old, the feeble and the infirm, but also investigations of non-cognitive mental phenomena (such as the emotions), of the possibility of cognition in other animals, and of the prospects for machine mentality, about which she declares "[she] will have nothing to say" (von Eckhardt, 1993, p. 6).

von Eckhardt defends her position by observing that no one would want to deny that adult, normal, typical cognition (ANTCOG, for short) is central to cognitive science. But it is not apparent why she should bother. Having defined "cognition" as computation across representations as a matter of stipulation, it follows that whatever *lacks* the capacity for computation across representations by definition *lacks* cognition. Her conception is analytic and immune from refutation by any evidence. Moreover, since she does bother, von Eckhardt becomes vulnerable to the complaint that she appears to be practicing (what might be described as) the method of selection and elimination, selecting the evidence that agrees with a predetermined conclusion and eliminating the rest (Fetzer, 1993a).

The practice of insulating your view from evidential refutation by rendering it true by definition, of course, is hardly new, and the method of selection and elimination has been practiced by attorneys and lesser sophists, no doubt, since humans first acquired language. But philosophers should not be so easily taken in. In its application to number theory, for example, the method of selection and elimination could be employed to “prove” that every number is even; applied to *Homo sapiens*, to “prove” that every human is male; applied to tosses of coins, to “prove” that every toss comes up tails. No one, after all, would want to deny that the study of even numbers is central to number theory, or that the study of males is central to anthropology, or the study of tails to tosses of coins.

That this is hardly the stuff of serious inquiry becomes apparent in light of (what is known as) *the requirement of total evidence*, according to which acceptable scientific conclusions must be based upon all of the available relevant evidence (Hempel, 1965). Evidence is relevant, in turn, when it makes a difference to the truth or falsity of an hypothesis (Fetzer, 1981). The existence of odd numbers is relevant to the hypothesis that all numbers are even, the existence of females to the hypothesis that every human is male, and the occurrence of heads to the hypothesis that every toss comes up tails. The study of the young and the old, feeble and infirm, animal cognition and machine mentality are likewise relevant to the hypothesis that cognition is computation across representations.

Unless, of course, we already knew that the young and the old, the feeble and the infirm, animals and machines were incapable of cognition. That would be a difficult claim to justify for anyone who eschews the study of these things, and von Eckhard does not make it. Her stance appears to be that understanding adult, normal, typical cognition is central to cognitive science and that, once it has been adequately understood, our attention should turn to things of other kinds. The only evidence relevant to adult, normal, typical cognition, thus understood, therefore, is evidence about adult, normal, typical cognition. This is the evidence that makes a difference to the conception of cognition as computation across representations for typical, normal, adult human beings, in which case, perhaps, she has not violated the requirement of total evidence, after all.

Computers and Cognition

Since the concept of cognition as computation across representations seems to suit computers, von Eckhardt’s definition of “computer” is of special interest:

> A computer is a device capable of automatically inputting, storring, manipulating, and outputting information in virtue of inputting, storring, manipulating and outputting representations of that information. (von Eckhardt, 1993, p.114)

She adds that these information processes are governed by a finite set of rules that are effective and are, in some sense, in the machine itself. Ultimately, she distinguishes between *conventional* and *connectionist* machines, where the former input, store, manipulate and output nodal representations, while the latter input, store, manipulate and output representations that are distributed instead.

More significantly, von Eckhardt invites attention to a subtle ambiguity in the use of the terms "data", "information", and even "input" and "output" within computer science, which are sometimes used to refer to *representations*—as marks, symbols, or whatever—and sometimes used to refer to *what those representations stand for*—as the information they convey (von Eckhardt, 1993, p. 113-114). Thus, to the extent to which computers input, store, manipulate and output representations—which appears to be the most appropriate conception—her definition of "computer" requires modification to reflect the difference at issue:

> A computer is a device capable of automatically inputting, storring, manipulating, and outputting representations of information in virtue of inputting, storring, manipulating and outputting other representations of information.

The difference between these formulations turns out to be an important issue. Other theoreticians have conceptions of computers that are less ambiguous.

Allen Newell and Herbert Simon (1976), for example, define "physical symbol systems" as systems that operate on symbols, which are physical patterns that can be combined to create other entities called "symbol structures". These are concatenations of symbols related by their relative sizes, shapes and locations:

> Besides these structures, the system also contains a collection of processes that operate on expressions to produce other expressions: processes of creation, modification, reproduction, and destruction. A physical symbol system is [therefore] a machine that produces through time an evolving collection of symbol structures. (Newell & Simon, 1976, p. 40)

It should be evident, of course, that Newell and Simon's conception might be reformulated or paraphrased in terms of "representations of information", where a physical symbol system is a machine that produces through time an evolving collection of representations of information, in the sense von Eckhardt intends.

An interesting difference between their views emerges from the emphasis that Newell and Simon place upon *computer commands*. Thus, for example, an expression "designates" an object when, given that expression (as input, presumably), a computer system can causally affect those objects or else behave in ways that depend upon those objects, where the implicit conception is one according to which input-expressions can be causally influential in bringing about the creation, modification, reproduction and destruction of symbol structures as output; or, in von Eckhardt’s language, input-expressions can be causally influential in bringing about inputting, storring, manipulating and outputting representations of information. Her emphasis seems to be upon data, their’s upon the manipulation of data.

Newell and Simon’s use of the terms “symbol” and “symbol structure” to stand for physical patterns that can be related by their relative sizes, shapes, and locations, moreover, strongly suggests a purely physical conception that harmonizes with the conception of computers as systems that have the capacity to manipulate representations. Indeed, an even more unambiguous endorsement of this notion may be found in John Haugeland’s conception of computers as “automatic formal systems”, which are devices (such as machines) that can automatically manipulate the tokens of formal systems acccording to the rules of those systems (Haugeland, 1981, p.10). These “tokens” are merely physical marks that can be distinguished by their relative sizes, shapes, and locations, where these marks can be infused with meaning when they are supplied with an appropriate interpretation.

Automatic Formal Systems

The conception of computers that emerges from these considerations is that computers are systems that are capable of inputting, storring, manipulating and outputting marks that are or can function as representations of information, but where the information they convey depends upon or presupposes the existence of an interpretation, interpreter, or mind. Haugeland’s conception of computers as automatic formal systems may be the most explicit on this point, because he readily concedes the difference between viewing “tokens” or “strings of tokens” as *meaningless markers*, which are manipulated according to the rules of some “self-contained game”, and viewing them as *meaningful strings* that bear some potentially significant relationship to the outside world (Haugeland, 1981, p. 22).

His remarks resonate with Newell and Simon’s observation that physical symbol systems exist within a world of objects that is larger than symbols and symbol structures themselves (Newell & Simon, 1976, 40). While they may emphasize the function of expressions as commands that control the creation, modification, reproduction, and destruction of things of these kinds, they clearly envision symbols and symbol structures as kinds of things that can be used to refer to other things beyond those systems themselves. Indeed, they ultimately propose (what they call) *the physical symbol system hypothesis* as a theory of mentality, according to which “A physical symbol system has the necessary and sufficient means for general intelligent action” (Newell & Simon, 1976, p.41).

At this point, the ambiguity to which von Eckhardt invites attention seems to manifest itself. Newell and Simon suggest their notion of “intelligence” has the same scope of intelligence as we see in human action: that in any real situation, behavior appropriate to the ends of the system and adaptive to the environment can occur, within some limits of speed and complexity. (Newell & Simon, 1976, p.41-42) A contention of this kind, however, would be plausible only if computers, for example, as physical symbol systems, have both motives and beliefs, insofar as a system with no motives has no “ends” to be achieved and a system with-no beliefs has no conception of the environment to guide it. If physical symbol systems *are* automatic formal systems—as mark-manipulating or as string-processing systems, in Haugeland’s sense—then they have to be supplied with an interpretation in order for those marks and those strings to have meaning.

Haugeland confronts this difficulty directly, by endorsing the supplemental conception of “semantic engines” as *automatic formal systems with interpretations* such that the meaning or the semantics of those marks and strings that are subject to inputting, storring, manipulating, and outputting “takes care of itself” (Haugeland, 1981, p.24), a notion that he attributes to Dennett (1981). The most obvious method by means of which it might be insured that marks are manipulated or strings are processed in such a fashion as to preserve an appropriate interpretation between them occurs when they are deliberately designed to sustain that interpretation. The most obvious reason automatic formal systems *can be* semantic engines is because we build them that way.

The conception of mark-manipulating and string-processing systems as semantic engines makes sense when we consider the possibility that those who build them impose interpretations upon software that motivate their design. Even the conception of computers as devices that input, store, manipulate and output information by virtue of inputting, storring, manipulating and outputting representatives of information makes sense so long as we understand that they convey information because we interpret them that way. But the hypothesis that physical symbol systems satisfying necessary and sufficient conditions for intelligence makes sense only if computers are capable of imposing interpretations upon the marks or strings they manipulate.

The Intentional Stance

Some theoreticians have supposed that machines do not actually have to have motives and beliefs to qualify as "semantic engines", when we are willing to treat them *as if* they have those qualities. The most striking advocate of this approach is Daniel Dennett, who advocates (what he calls) "the intentional stance" (Dennett, 1971). According to him, the decision is pragmatical:

> Lingering doubts about whether [a] chess-playing computer really has beliefs and desires are misplaced; for the definition of intentional system I have given does not say that intentional systems really have beliefs and desires, but that one can explain and predict their behavior by ascribing beliefs and desires to them... . The decision to adopt the strategy is pragmatic, and is not intrinsically right or wrong. (Dennett, 1971, p.224-225)

The advantage of adopting the intentional stance, he claims, is that "it is much easier to decide whether a machine can be an intentional system than it is to decide whether a machine can really think, or be conscious, or morally responsible" (Dennett, 1971, p. 235), a point about which there ought to be no doubt.

There are far simpler pastimes than philosophy, where some questions are hard. If what we want to know is whether or not chess-playing computers, human beings and other things really can *think* or really have *beliefs* and *desires*, then Dennett's position provides no answers. Indeed, it is not hard to establish that the intentional stance promises far more than it can deliver. Explanations, after all, are adequate only when they happen to be true, while the intentional stance eschews questions of truth and falsity about the actual mental states of intentional systems. If human beings can think and have motives and beliefs, their behavior can be explained (at least, in part) by ascribing them; if chess-playing computers cannot think and have no motives or beliefs, their behavior cannot be explained merely by treating them as if they did (Fetzer, 1990a, p.14).

In his book, *Kinds of Minds* (1996), Dennett persists in defense of the indefensible. In an attempt to respond to critics who have found the implications for evolution of his *Consciousness Explained* (1991) and *Darwin's Dangerous Idea* (1995) "dubious or even alarming" (Dennett, 1996, p.173), he reiterates the range of things whose behavior is alleged to be capable of successful explanation and prediction from this point of view, which, he claims, includes "self-replicating molecules, thermostats, amoebas, plants, rats, bats, people, and chess-playing computers". Thus, the crucial move that warrants predictions is making the assumption that they are "smart agents": "This bold leap of supposing that the agent will make only the smart moves (given its limited perspective) is what gives us the leverage” to successfully predict their behavior (Dennett, 1996, p.34).

We have already discovered that the intentional stance cannot support explanations of the behavior of any systems that lack the properties it would ascribe to them. It should therefore be viewed at best as no more than an instrumental device for deriving predictions. But while Dennett describes the assumption that an entity will make only the *smart moves* (given its limited perspective), as a “bold leap”, he subsequently reports—in the very same volume—that “experiments with animals demonstrate surprisingly stupid streaks—almost unbelievable gaps in the animals’ knowledge of their own environments” (Dennett, 1996, p.91). His “bold leap” at least sometimes would result in massive blunders, which implies that the intentional stance can be as irrelevant for deriving predictions as it is for deriving explanations. It applies to predict the behavior of agents only if they are smart.

The intentional stance offers nothing that alters the conception of computers as mark-manipulating and as string-processing systems, which tend to qualify as semantic engines because they have been designed with interpretations in mind— that is, in the minds of the designers rather than in the minds of those machines. When marks are manipulated and strings are processed in accordance with rules that correspond to an interpretation that might be imposed upon them by their users, moreover, then those mark-manipulating and string-processing systems not only qualify as automatic formal systems but as syntax-processing systems, where *syntax* consists of marks and sequences of marks that are capable of sustaining systematic interpretations by virtue of which those marks and sequences become representations that stand for information for the users of those systems.

Syntax and Semantics

A more promising and less inadequate defense of computer mentality has been advanced by William J. Rapaport (1988), who has attempted to explain how semantics can be dervied from syntax even apart from an interpretation, interpreter, or mind. Rapaport contends that sufficiently complex networks of nodes can implement semantics, where the crucial ingredient of his position is reliance upon (what he refers to as) “Internal Semantic Networks”, which are networks of interconnecting arcs that meet at nodes. Thus, Rapaport contends,

> All of this is what I shall call *internal* semantics: semantics as an interconnected network of internal

> symbols—a “semantic network” of symbols in the “mind” or “knowledge base” of an intelligent system, artificial or otherwise. The *meaning* of “1” or of any other symbol or expression, is determined by its locus in this network ... as *well as* by the way it is *used* by various processes that reason using the network. (Rapaport, 1988, p.94)

Rapaport’s position, however, appears to be a sophisticated version of the ambiguity between representations of information and information itself to which von Eckhardt invited attention, which was manifest in Newell and Simon’s conception.

The problem can be illustrated by means of an ordinary dictionary. Some of the words appearing there (the *definienda*) are defined by means of other words (their *definiens*), where those words in turn are defined by means of other words, sometimes generating circular definitions. These definitional chains might be no> more than meaningless sequences of marks connected to other meaningless sequences of marks, where sequence “xyz” is related to “zab”, which in turn is related to “bcd”, and so on back to “xyz”. The existence of connections of this kind, after all, does not mean that any of these sequences are meaningful, even if they are designated as “definiendum” and “definiens”, as “words” and “meanings”, as “syntax” and “semantics”. They might all be meaningless marks (Fetzer, 1991, p.44).

Even if computers are physical symbol systems in Newell and Simon’s sense, for example, that does not infuse the symbols and symbol structures they manipulate with meaning. Their conceptions of symbols and symbol systems as physical sequences that are distinguishable on the basis of their sizes, shapes and relative locations surely does not make them meaningful. The physical symbol system hypothesis thus equivocates between “physical symbol systems” in the weak sense of systems that manipulate physical sequences distinguishable on the basis of their sizes, shapes and relative locations, and “physical symbol systems” in the strong sense of systems that manipulate physical sequences that are distinguishable on the basis of their physical properties but which are infused with meaning.

Rapaport’s conception thus appears to fall prey to a similar ambiguity, even if it does so in a less obvious and more sophisticated form. The most adequate analysis of those surveyed thus far still seems to be Haugeland’s conception of computers as automatic formal systems, which input, store, manipulate and output possibly meaningless marks, where these mark-manipulating and string-processing systems may often qualify as semantic engines because they have been designed with interpretations in mind. They are thus converted from mark-manipulating systems into syntax-processing systems, where the marks and mark-sequences they manipulate are capable of sustained systematic interpretations. To the extent to which complexity of syntax affects interpretations that can be systematically imposed upon it, Rapaport's conception is well-founded. But the existence of interpretations nonetheless presuppposes the existence of interpreters or minds.

Indeed, this conception of computers appears consistent with John Searle's recent work, according to which syntax is not intrinsic to physics (Searle, 1992). Although computation is commonly defined syntactically in terms of the manipulation of symbols, of symbol structures, and of representations of information, which are differentiated on the basis of their sizes, shapes and relative locations, those conceptions presuppose that the marks and sequences of marks that are subject to manipulation have the capacity to sustain an interpretation and are suitable to serve that function: "This has the consequence that computation is not discovered in the physics, it is assigned. Certain physical phenomena are used or programmed or interpreted syntactically" (Searle, 1992, p.225). Thus, from this point of view, computing is not a meaningful activity for computers.

Kinds of Signs

Although that may sound paradoxical, computers, thus understood, appear to be devices that can be used to perform computations, where those computations possess no meaning for those devices. This makes sense from the perspective of the theory of signs advanced by Charles S. Peirce (Hartshorne & Weiss, 1960). According to Peice, a *sign* is a something that stands for something else in some respect or other for somebody. For something to stand for something thus implies a tridaic relation between a sign, a something, and a somebody for whom that sign stands for that something in some respect or other. The most obvious reason why computations may be meaningless for computers is that the marks or strings that they manipulate simply do not stand for anything else for them.

Peirce identified three kinds of signs on the basis of their semantic grounding relations, first, those that resemble those other things, which he called *icons*; second, those that are causes or effects of those other things, which he called *indices*; third, those that are habitually associated with those other things, which he called *symbols*. Icons thus include photographs, paintings, and statues, for example, at least to the extent to which they resemble what they stand for. Indices as causes or effects of that for which they stand include fire in relation to smoke, ashes in relation to fire, and smoke and fire in relation to damage to life and limb. Symbols, which are merely habitually associated with that for which they stand, thus include the words and sentences of various natural languages, where the institutionalization of those associations turns them into conventions.

The ambiguity between representations and information to which von Eckhardt invites attention receives more confirmation here. Even if computations have no meaning for computers, computers can still be devices that are or can be used to perform computations by their users, who infuse those marks and strings with meaning by supplying interpretations that they can systematically satisfy, just as Haugeland’s conception of “automatic formal systems” requires. Computers appear to be automated formal systems that can become “semantic engines” when they are infused with meaning by their users. The marks and strings that they manipulate can function as syntax relative to interpretations imposed by interpreters or minds that function as their counterpart semantics.

The objection could be raised that computers operate on the basis of artificial languages, such as Pascal, LISP, and Prolog, just as humans operate on the basis of natural languages, such as English, French, and German. These “high-level” programming languages, of course, are connected by means of interpreters or compilers to machine languages, written in binary code, which thereby implement computer commands in forms that computers can execute directly, while “linguistic” mental states are implemented by means of neurons, ganglions and synaptic connections to speech and other behavior activators. Even if natural languages employ words and sentences that are only habitually or conventionally associated with that for which they stand, nothing has shown that the same is not true for computers in relation to programming languages.

Moreover, Newell and Simon’s conception of computers as physical symbol systems is especially pertinent here, insofar as they emphasize the arbitrary character of the connections between “symbols” and the operations for which they stand (Newell & Simon, 1976, p.41). There thus appears to be at least some kind of conventional connection between physical symbols that activate the behavior of *symbol systems* in Newell and Simon’s sense and the physical symbols that activate the behavior of *symbol-using systems* in Peirce’s sense. Systems of both kinds appear to be causal systems, in the sense that their behavior at time *t2* appears to be brought about by complete sets of relevant conditions at time *t1* and the principles that govern their behavior. The difference between them thus appears to be the kinds of causal system they happen to be.

The Stored Program

The conception of computers as automatic formal systems incorporates one of the distinguishing characteristics of (what are known as) von Neumann machines, which is the *stored program*. A “stored program” consists of a set of instructions written in a programming language, which allows a computer to perform operations without the necessity for human intervention. It is this property that makes these formal systems “automatic”. Although today computers are assumed to be von Neumann machines, certain properties of computers can be illustrated in relation to other devices that are not “automatic” in this sense but which nevertheless are or can be used to perform computations by their users, where their users provide their own programs and automation to those devices.

The abacus supplies a perfect illustration. An abacus is a device consisting of balls or beads strung on wires or rods set in a frame, which is typically made of wood but sometimes of other material. Although these strings of beads have no intrinsic meaning, they can be interpreted as standing for numbers and numerical relations. Thus, each group of beads can be taken to stand for classes of numbers, such as digits, 10s, 100s, and so forth, where by moving the beads a skilled operator can add, subtract, multiply and divide, in some cases extremely rapidly. The operator, in this case, of course, must move them in accordance with suitable instructions, which implements the automation of a program for numerical operations of this kind. The abacus is thus a device whereby computation can be performed, where an abacus combined with its user qualifies as a “semantic engine”.

Thus, I am suggesting that greater insight about the nature of computers can be secured by viewing from the perspective of the abacus. That is, we may gain a greater degree of understanding about the character of contemporary (digital) machines when we think of them as *a special kind of abacus*. Admittedly, there is the important difference between computers that can execute numerical operations and those that can execute symbolic operations by virtue of the capacity to sustain alphanumeric as well as numeric interpretations. But that is merely a refinement to which we will return. The point is that a sophisticated abacus equipped with a stored program—but without its user—could qualify as a physical symbol system or as an automatic formal system. But it could not qualify as a semantic engine or even as a machine that manipulates representations of information.

The objection that computers operate on the basis of artificial languages that relate computer programs to computer behavior by means of interpreters and compilers, while humans operate on the basis of natural languages that connect mental states to their behavior by means of neurons, gangions, and synapses also turns out to be misleading. Even if words and sentences in natural languages are only habitually or conventionally associated with that for which they stand, the purely internal comparison disregards a crucial difference, because a causal factor in the operation of a mind is that something activates neurons, ganglions and synapses leading to speech and other behavior *because it stands for something else for that system*. Symbols used by humans are signs in Perice’s sense.

The observation could be posed, no doubt, that even if the arguments that I have advanced demonstrate that ordinary (digital) computers, including those that are equipped with stored programs, are not thinking things, that does not demonstrate that thinking things are not even more sophisticated kinds of ordinary (digital) machines. In other words, suppose we now understand computers to be mark-manipulating and string-processing systems where the manipulation of those marks and strings can be governed by programs enabling them to be interpreted as syntactical tokens and as syntactical strings that can sustain systematic semantic interpretations. That demonstrates that ordinary (digital) computers are not thinking things. But it does not demonstrate that thinking things are not ordinary (digital) computers, say, with interpreters of their own.

Semiotic Systems

This rejoinder may be especially appealing to those who are inclined to embrace the computational paradigm, because it suggests that they have not been all wrong after all but instead might be at least half-right. It also harmonizes almost exactly with Haugeland’s conception of “semantic engines” as automatic formal systems that can sustain systematic semantic interpretation—when that interpretation is provided by that system itself! Even Haugeland acknowledges several possible ways in which the computational conception might need supplementation to yield a complete account of the nature of thinking things, including consciousness, intentionality, and capacity to care (Haugeland, 1981, p.32-24).

Thus, from this perspective, if the satisfaction of any of these conditions could be combined with the computational conception to provide an adequate account, then the computational conception, which seems to be insufficient for mentality, might nevertheless turn out to be necessary. One of the most promising of these possibilities concerns *intentionality*, where Haugeland acknowledges a traditional distiction between two kinds of intentionality, one “original”, the other “derived”:

> The idea is that a semantic engine’s tokens only have meaning because we give it to them; their intentionality, like that of smoke signals and writing, is essentially borrowed, hence *derivative*. To put it bluntly: computers themselves don’t mean anything by their tokens (any more than books do)—they only mean what we say they do. Genuine understanding, on the other hand, is intentional “in its own right” and not derivative from anything else. (Haugeland, 1981, p.33)

The problem this raises, as Haugeland observes, is to identify precisely what may be required for a system to possess original intentionality and be a thinking thing.

The difference between symbol systems in Newell and Simon’s sense and symbol-using systems in Peirce’s sense appears to matter here. “Symbols” in Peirce’s sense are signs that stand for something else in some respect or other for somebody, who is the user of those symbols. Thus, symbols as signs in Peirce’s sense must be meaningful for their users. “Symbols” in Newell and Simon’s sense are merely physical tokens, which can be distinguished and manipulated on the basis of their sizes, shapes and relative locations by suitably programmed systems, but do not have be meaningful to the systems that use them. The fundamental difference between them thus appears to be a distinction between “signs” that are meaningful for the systems that use them and “signs” that are only meaningful for a system’s users. One displays original intentionality, the other derivative.

When consideration shifts from signs themselves to sign users, however, an alternative conception of mentality emerges, namely: the conception of *minds as sign-using (or “semiotic”) systems* (Fetzer, 1988, 1989, 1990a, 1991). Since there are three different kinds of signs, there are three corresponding kinds of minds, namely: Type I minds, which possess iconic mentality; Type II minds, which possess indexical mentality; and Type III minds, which possess symbolic mentality. There even seem to be at least two stronger kinds of minds, capable of reasoning and of criticism. This approach, moreover, encompasses the use of signs other than symbols. Many varieties of human thinking, including dreams, daydreams and ordinary thought, involve the use of images or associations predicated upon resemblance relations and causal connections (Fetzer, 1994b, 1998).

To provide a conception that applies to humans without precluding the prospect that other animals and even machines might possess mentality, the phrase, “for somebody”, ought to be displaced by “for something”, in Peirce’s definition; thus, a sign is something that stands for something else in some respect or other *for something*, which might be human, (other) animal or even machine. Indeed, insofar as these are successively stronger and stronger kinds of mentality, the account of minds as semiotic systems implies that lower species should possess lower kinds of mentality and higher species higher. *Homo sapiens* thus appears to be distinguished, not by virtue of being the only species possessing mentality, but by possessing the highest species of mentality (Fetzer, 1993b,1994b, 1996).

Consciousness and Cognition

The conception of minds as semiotic (or “sign-using”) systems promises to provide solutions for problems about consciousness and cognition. It implies a conception of *consciousness* according to which a system is conscious (with respect to signs of specific kinds) when it has the (inherited or acquited) ability to utilize signs of that kind and is not incapacitated from the exercise of that ability. And *cognition* occurs as the outcome of a causal interaction between signs of specific kinds within a suitable causal proximity and systems that are conscious with respect to signs of that kind (Fetzer, 1989, 1990a, 1991). Thus, successful communication tends to occur between two systems when the same signs possess the same *meaning* for them both, even though their actual behavior—their speech and other motor activity—may differ from case to case.

The differences in actual behavior that different systems manifest tends to arises from difference in the complete sets of relevant conditions affecting their behavior, which, for humans, include their motives, beliefs, ethics, abilities, and capabilities. Different instances of signs would have the same meaning for sign users if their potential behavior—their dispositions to behave in various ways—were the same across every possible context (Fetzer, 1989, 1991, 1993c). Since some dispositions to behave one way or another may be probabilistic, however, it does not follow that various sign users in the very same contexts, when exposed to the very same signs, would therefore display the same behavior. Such an outcome would be expected only if the relevant dispositions were deterministic.

This conception thus exemplifies Peirce’s observation that “the most perfect account of a concept that words can convey” would be a description of the kinds of behavior to which it gives rise in relation to causally-related states, including preexisting motives and beliefs (Fetzer, 1989, pp. 33-34). A licensed driver who knows the rules of the road and is not incapacitated from properly driving a car might run a stop sign because it was obscured by overhanging shrubbery, even though that was not his intention; but another driver who also know the rules of the road might also run a stop sign, not because it was obstructed but because he was intoxicated; and another driver might run a stop sign to rush his wife to the hospital because of the onset of labor. Yet they might all still understand that a stop sign *means* coming to a complete halt and proceeding when it is safe to do so.

The crucial factor that differentiates between symbol systems, automatic formal systems and other devices useful for computation, and semiotic systems thus appears to be that semiotic systems, but not devices of these other kinds, actually take things to stand for other things in some respect or other. What makes a system “semiotic” thus becomes that the behavior of the system is causally affected by the presence of a sign *because* that sign stands for something else iconically, indexically, or symbolically, for that system. Those things for which signs stand, moreover, may include abstract, theoretical, non-observable, or non-existent objects and properties, which may be incapable of exerting any causal influence on a system themselves. Thus, while von Eckhardt suggests internalistic conceptions of the meaning of representations for conventional and connectionist systems, she misses this crucial element (von Eckhardt, 1993, p.292-297; Fetzer, 1990, Ch. 9).

It should also be observed that the semiotic conception supports the capacity to make a mistake as a *criterion of mentality*, where a “criterion” in this sense is a usually reliable evidential indicator of the presence or absence of the property under consideration. Mistakes occur when something takes something to stand for something other than that for which it stands, which is the right result (Fetzer, 1988). In defense of dispositional conceptions of meaning, moreover, Paul Coates (1997) has remarked that cognition involves both *intensions* (with an “s”) and *intentions* (with a “t”). When we are in standard, truth-seeking contexts, for example, we may accept conclusions, even though they make us uncomfortable or are threatening; otherwise, we may not. Unless computers are capable of forming intentions, their behavior can be neither meaningful nor purposive. Whatever computers can do, therefore, it does not appear to be anything of this kind.

Turing Machines

Other lines of argument yield similar conclusions. The nature of computation has classically been defined by reference to a Turing machine, which consist of an infinitely long, segmented tape and a device that can perform four operations on such a tape, namely: make a mark, erase a mark, move one segment forward, and move one segment backward. Classic theorems by Church and Turing have shown that every computable problem can be performed by a Turning machine. Indeed, as Daniel Cohen has observed, the notion of a Turing machine itself has made the notion of computability (more or less) precise:

> It is believed that there are no functions that can be defined by humans, whose calculation can be described by any well defined algorithm that people can be taught to perform, that cannot be computed by Turing machines. The Turing machines is believed to be the ultimate calculating mechanism. (Cohen, 1986, p.790; original italics).

Yet even the notion of a Turing machine seems to be afflicted with ambiguity.

As John Kearns (1997) has remarked, Turing machines are definable in at least two ways . In the first sense, a Turing machine is viewed as a mark-manipulation procedure for someone to carry out. In the second sense, a Turing machine is a device that operates independently of human intervention. When we view the machine as a mark-manipulation procedure, a description of the machine has the character of a (possibly complex) sequence of rules for a person to carry out, which s/he may or may not be actually capable of doing. When we view the machine as an independent device, however, a Turing machine is a spatio-temporal mechanism operating by means of causal principles.

From the perspective of the account of minds as semiotic sytems, perhaps the most important ramification of this differentiation is that, while Turing machines in the first sense possess intensionality (with an “s”) and intentionality (with a “t”), Turing machines in the second sense possess neither. For a person to follow a (possibly complex) sequence of rules they are supposed to carry out, they would presumably have to have the intention of doing so (if their success, such as it may be, were to qualify as anything but luck) and would need to understand at least the kinds of operators and operands on which they were performing (for similar—indeed, if not identical—good reasons). A spatio-temporal device of the corresponding kind can do without either intensions or intentions.

Ultimately, Kearns’ position requires qualification, because he wants to draw a strong distinction between systems capable of intentionality and others that are not, where those systems that are capable of intentionality are not “causal”. Systems of both kinds appear to be causal systems, in the sense that their behavior at time *t2* appears to be brought about by complete sets of relevant conditions at time *t1* and the principles that govern their behavior. The difference between them thus appears to be the kinds of causal system they happen to be. A person attempting to follow a (possibly complex) sequence of rules as described would qualify as a semiotic system, while the counterpart device would not.

A recent but related controversy has emerged over the scope and limits of the Church-Turing thesis, precipitated by a distinction between different conceptions of its intended domain. Carol Cleland (1993), in particular, has urged that the application of algorithms to computable numerical functions does not exhaust the class of effective decision procedures due to the existence of (what she refers to as) *mundane procedures*, such as following recipes for cooking, executing directions for driving, and similar activities (Horsten and Roelants 1995, Cleland 1995). Without dissecting the issue in detail, distinctions can surely be drawn between (purely) numerical, alphanumerical and other kinds of procedures. Cleland's mundane procedures seem to be a class of procedures that are not Turing computable, but are effective, due to laws of nature (Fetzer, 1990b).

Mental Algorithms

The final line of defense can be marshalled by asserting that thought processes, at the very least, are governed by mental algorithms. Even if Turing machines and computing devices generally succumb to differences between original and derivative intentionality as symbol rather than semiotic systems, surely semiotic systems nevertheless satisfy the conditions for being algorithmic in the sense that thought processes have to have the sequential structure of effective procedures. This is the sense in which Haugeland suggests that thinking is reasoning, reasoning is reckoning, reckoning is computation, and computation is cognition, where the boundaries of computability define the boundaries of thought. While it sounds appealing, this argument will not do.

The crucial issue become the nature of mental algorithms, which hangs on the nature of algorithms themselves. Virtually every contributor to the field agrees that, as effective decision procedures, algorithms must be sequential, completable, definite, and reliable. Alonzo Church, Stephen Kleene, and John von Neumann independently substantiate that characterization (Church, 1959; Kleene, 1967; and von Neumann, 1958), which von Eckhardt incorporates in passing by means of her observation that the information processes that computers perform are governed by finite sets of effective rules that are, in some sense, in the machine itself. The very idea of "executing an algorithm", moreover, appears to be fundamental to the conception of computer science as the science of solving problems with machines. Algorithms are solutions.

Having already suggested that dreams, daydreams, and ordinary thought processes are not algorithmic, an illustration appears to be appropriate. Not long ago, looking out a window of my home, which is located in Duluth and is situated on bluffs overlooking Lake Superior, I noticed the house was enveloped with fog. That reminded me of a postcard that depicted nothing but fog with only the name, "Seattle", across the bottom. That made me think of my son, Bret, who has lived in Seattle for years, and about plays, of which he has written several. I then thought of his first play, which both I and one of my brothers, who graduated from Santa Cruz, had read before seeing it performed. We both thought it was a tragedy, but it played as an hilarious comedy. My wife's sister thought it was the most depressing thing she had ever read.

This train of thought can easily be understood as a semiotic phenomenon. The transition from observing the fog to the postcard, of course, was caused by an iconic relation of resemblance. The postcard said, "Seattle", which is a symbol for a city. Thinking of the city brought to mind my son, whom I have visited there, as an habitual association. Thinking of his productivity in writing plays, which is an indexical connection, I thought of the first play that he wrote, as the first outcome of his efforts. The causal consequences of that play for my brother and me were brought to mind, especially how impressed we were when we finally had the chance to see it performed. The effect was sensational. And how comic it seemed by comparison to the effect it had induced on Donna Gire.

This train of thought appears to be typical, normal, adult, human cognition. It is not algorithmic. There is no completable, definite, and reliable effective decision procedure that underlies these causal transitions, which appear to be influenced at least as much by past experience and emotional connotations as they are by logic and reasoning. No reckoning at all is involved. There is nothing here to support the inference that thought processes are governed by finite sets of effective rules. And this is not peculiar to thinking of this specific kind. Even perception and memory fail to qualify as activities that are governed by mental algorithms. Wanting to perceive something and trying to remember something are certainly problems we encounter in life. But they are problems for which life, alas, provides no effective solutions (Fetzer, 1994a, 1998).

Special Kinds of Signs

A computer thus appears to be a sign of a special kind, namely: a sign or a system of signs of the kind that can facilitate computations, where "computations" involve processing information by processing representatives of that information by means of finite sets of effective rules, at least some of which may take advantage of laws of nature as their theoretical foundation. Indeed, there seem to be at least *three kinds of* *computers*, namely: iconic, indexical, and symbolic, which qualify as (possible or actual) computers because of the existence of resemblance relations (for iconic computers), of cause-and-effect relations (for indexical computers) or of conventional relations (for symbolic computers) between their semiotic parts and the things for which they stand.

Examples of iconic computers seem to include models, maps and diagrams. These instantiate arrangements of objects that can exhibit semantic grounding relations—such as Cleland’s “mirroring relations” (Cleland, 1995)—by virtue of resembling what they stand for, where mathematical functions can relate the properties of those things to those of things for which they stand. Analogous examples of indexical and symbolic computers that can be employed for computational purposes include sundials, tree-rings and geological strata as effects of various causes, which can be related by computable functions to the causes that produced them; and abaci, slide-rulers, and Turing machines, which can instantiate mathematical functions that relate things on conventional grounds.

At least one more ambiguity thus arises in discussing computers and computation, since the existence of a *possible* computer implies the existence of a possibly unnoticed semantic grounding relation between something and something else. In this sense, a computer can exist even in the absence of any interpretation, interpreter, or mind. The existence of an *actual* computer, however, not only implies the existence of an actual grounding relation of that kind (since every actual computer is a possible computer), but also implies the existence of something that has noticed the existence of that relationship and thus the existence of a corresponding interpretation, interpreter, or mind. This distinction
may explain the rationale of those who claim that anything can be a computer.

The existence of mundane procedures, however, suggests that there can be effective decision procedures that are not Turing computable. And the existence of thought processes that are non-algorithmic further substantiates the view that the computational paradigm cannot be sustained. Indeed, at best, it appears to be no more than an overgeneralization based upon what seems to be a special kind of thought process. As Searle has observed, “(N)othing is intrinsically computational, except of course conscious agents intentionally going through computations” (Searle, 1992, p. 225). Even though a Turing machine with a stored program may be able to perform operations without the necessity for human *intervention*, that does not imply that such a device can perform meaningful operations without the benefit of human *interpretation*.

To the extent to which cognition as computation across representations is supposed to be a purposive, meaningful, algorithmic, problem-solving activity, computers seem to be incapable of cognition. The marks and strings they manipulate have no meaning for them, and they are incapable of purposive actions: they lack intensions (with an "s") and intentions (with a "t"). They are devices that can facilitate computations on the basis of semantic grounding relations in those cases in which they exist. Even their algorithmic, problem-solving character arises from their interpretation by human users. In the strict sense, computers as such—apart from their users—are not only incapable of cognition, but even incapable of computation, properly construed. If we want to understand the nature of thought, we are going to have to study thinking, not computing.

Even adult, normal, typical cognition as displayed in ordinary thought is neither algorithmic nor problem-solving and therefore fails to satisfy conditions that computationalists would impose upon it. The ANTCOG conception thus appears to be a subterfuge for stipulating the definition of "cognition", which ignores (what seems to be) abundant evidence that disconfirms or falsifies the prevailing paradigm. The requirement of total evidence insists that this evidence be taken into account. Whether anomalies of these and other kinds will dislodge the computational conception remains to be seen, where the history of science is not overly reassuring. But at least this much should be obvious. The conception of minds as semiotic systems supplies an alternative paradigm for understanding the nature of mental phenomena—one that appears to be capable of overcoming anomalies that computationalism cannot.

References

Church, A. (1959). Logistic System. In D. Runes (Ed.) *Dictionary of Philosophy*. Littlefield, Adams & Company, pp. 182-183.

Cleland, C. (1993). Is the Church-Turing Thesis True? *Minds and Machines, 3*, 283-312.

Cleland, C. (1995). Effective Procedures and Computable Functions. *Minds and Machines, 5*, 9-23.

Coates, P. (1997). Meaning, Mistake and Miscalculation. *Minds and Machines, 7*, 171-197.

Cohen, D. (1986). *Introduction to Computer Theory*. John Wiley & Sons.

Dennet, D. (1971). Intentional Systems. *Journal of Philosophy, 68*, 87-106. Reprinted in Haugeland (1981), pp. 220-242. Page references are to the reprinted version.

Dennet, D. (1981). Three Kinds of Intentional Psychology. In R. A. Healey (Ed.), *Reduction, Time and Reality* (pp. 37-61). Cambridge, UK: Cambridge University Press.

Dennet, D. (1991). *Consciousness Explained*. Little, Brown & Co.

Dennet, D. (1995). *Darwin's Dangerous Idea*. Simon & Schuster.

Dennet, D. (1996). *Kinds of Minds*. Basic Books.

Fetzer, J. H. (1981). *Scientific Knowledge*. D. Reidel Publishing Company.

Fetzer, J. H. (1988). Signs and Minds: An Introduction to the Theory of Semiotic Systems, In J. Fetzer (Ed.), *Aspects of Artificial Intelligence* (pp. 13-161). Dordrecht, The Netherlands.

Fetzer, J. H. (1989). Language and Mentality: Computational, Representational, and Dispositional Conceptions. *Behaviorism, 17*, 21-39.

Fetzer, J. H. (1990a). *Artificial Intelligence: Its Scope and Limits*. Dordrecht, Kluwer Academic Publishers.

Fetzer, J. H. (1990b). Artificial Intelligence Meets David Hume: A Reply to Pat Hayes. *International Journal of Expert Systems, 3*, 239-247.

Fetzer, J. H. (1991). *Philosophy and Cognitive Science.* Paragon House Publishers.

Fetzer, J. H. (1993a). A Piece of My Mind: Lundberg, JFK and JAMA. *The Fourth Decade, 9*, 35-40.

Fetzer, J. H. (1993b). Review of Merlin Donald’s Origins of the Modern Mind, *Philosophical Psychology, 6*, 339-341.

Fetzer, J. H. (1993c). *Primitive Concepts: Habits, Conventions, and Laws, in Definitions and Definability* (Ed. J. Fetzer, pp. 51-68). Kluwer Academic Publishers.

Fetzer, J. H. (1994a). Mental Algorithms: Are Minds Computational Systems? *Pragmatics and Cognition, 2*, 1-29.

Fetzer, J. H. (1994b). Review of Donald Griffin’s Animal Mind. *Human Ethology Newsletter, 9*, 15-17.

Fetzer, J. H. (1996). Do Animals have Minds? Discussion Review: M. S. Dawkins, Through Our Eyes Only: The Search for Animal Consciousness. *The Journal of Social and Evolutionary Systems, 19*, 187-192.

Fetzer, J. H. (1998). People are Not Computers: (Most) Thought Processes are Not Computational Procedures. *Journal of Experimental and Theoretical AI, 10*, 371-391.

Hartshorne, P. & Weiss, P. (1960). *The Collected Papers of Charles S. Peirce* (Vols. 1 and 2). Harvard University Press.

Haugeland, J. (Ed.). (1981). *Mind Design: Philosophy, Psychology, Artificial Intelligence*. The MIT Press.

Haugeland, J. (1981). Semantic Engines: An Introduction to Mind Design, In J. Haugeland (Ed.) *Mind Design: Philosophy, Psychology, Artificial Intelligence* (pp. 1-34). The MIT Press.

Hempel, C. G. (1965). *Aspects of Scientific Explanation*. The Free Press.

Horsten, L. & Roelants,H. (1995). The Church-Turing Thesis and Effective Mundane Procedures. *Minds and Machines, 5*, 1-8.

Kearns, J. (1997). Thinking Machines: Some Fundamental Confusions. *Minds and Machines, 7*, 269-287.

Kleene, S. C. (1967). *Mathematical Logic*. John Wiley & Sons, 1967.

Newell, A. & Simon, H. (1979). Computer Science as Empirical Inquiry: Symbols and Search. *Communications of the ACM 19*, 113-126. Reprinted in Haugeland (1981), pp. 35-66. Page references are to the reprinted version.

Rapaport, W. J. (1988). Syntactic Semantics: Foundations of Computational Natural Language Understanding. In J. Fetzer, (Ed.), *Aspects of Artificial Intelligence* (pp. 81-131). Kluwer Academic Publishers.

Searle, J. (1992). *The Rediscovery of the Mind*. The MIT Press.

Stilling, N. A. et al. (1987). *Cognitive Science: An Introduction*. The MIT Press.

Thagard, P. (1995). Review of Barbara von Eckhardt, What is Cognitive Science? *Philosophy of Science, 62*, 345-346.

Von Eckhardt, B. (1993). What is Cognitive Science? The MIT Press.

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

- This chapter originally appeared in Minds and Machines 7/3 (August 1997), pp. 345-364. (By permission of the author)