The Monist
Volume 80
Number 3, July 1997
ARTICLES
Patrick J. Hayes: WHAT IS A COMPUTER? AN ELECTRONIC DISCUSSION

Patrick J. Hayes: WHAT IS A COMPUTER? AN ELECTRONIC DISCUSSION

1. Introduction

Page 389

An e-mail discussion can be rendered into print in several ways. Rather than trying to imitate a genuine conversation, this is a personal essay containing comments and replies by the other contributors. Most of the substantial points made in the e-mail discussion are contained here, although not always in the order they happened.

Page 389

The problem posed was to give a characterisation of a computer. According to Searle, the notions of "computer" and "computation" are not natural kinds, and anything with a sufficient number of states can be thought of as performing any computation, simply by interpreting its states to have appropriate meanings; this is always legitimate since such external interpretation is the only means by which the purely formal patterns in a conventional computer can be given meanings in any case. Thus a claim that something is a computer, or operates computationally, is simply vacuous. For example, a wall can be said to be running Wordstar, and our digestive systems have as much claim to being computational as our nervous systems do.

Page 389

Many agree that this conclusion seems evidently wrong, but it is not clear how to refute it. One approach, suggested by Valerie Hartcastle, is to allow Searle to have his logical point but to deny its claimed significance. Hayes suggests that an alternative way to characterise computers than the "abstract mechanism" used in conventional computability theory might avoid Searle's reductio ad trivium. Selmer Bringsjord disagrees and urges that the point simply needs some modal care; while it may be logically possible that anything be a computer, this is not physically possible. Istvan Berkeley raises an interesting issue by claiming that "computation" is inherently ambiguous. George McKee suggests that something is a computer when it is being programmed; but with a very broad notion of "programming"; and Rob Stufflebeam suggested the broadest definition of all: a computer is something that implements a function.

Page 389

One interesting disagreement concerned whether or not the human nervous system — the brain, in short — is a computer. While all the contributors agree that

Page 389

Page Break 390

it probably is one, Hayes takes this to be an empirical question, while Hartcastle and Berkeley want to make it matter of definition. This difference seems to reflect a well-known methodological split within cognitive science, between those who see the discipline as defined by a collection of ideas emerging from computer science, and those who regard it as defined by its subject matter, ie human and animal cognition. If history proves that the best explanation of human cognition is provided by quantum physics and microbiology, then from the first perspective cognitive science will have failed; from the second, it will simply have found a better theory. The natural move then for someone who insists that brains are computers would be to stretch the notion of "computer" to include such apparently noncomputational phenomena, as McKee suggests.

Page 390

I begin with a summary of the "target" article written by me to stimulate discussions.

2. Real computers

Page 390

First, I take it as simply obvious both that computers exist and that not everything is a computer, so that, contra Searle, the concept of "computer" is not vacuous.

Page 390

By "computer" I mean a machine which performs computations, or which computes. On this understanding, a Turing machine is not a computer, but a mathematical abstraction of a certain kind of computer.

Page 390

Computer science recognises many kinds of computer and computation, and we should try to include them all. For example, the notion of a "virtual machine" is fairly central, so an adequate account of computer must somehow account for virtual machines.

> [Istvan Berkeley:] There is a problem here, because it is perfectly possible to instantiate a Turing machine (not of course a universal Turing machine) on a regular Von Neumann machine as a virtual machine. But according to the above, this would make Turing machines computers, thereby contradicting the previous claim.
>
> (In reply, I would urge a distinction between a Turing machine itself — a mathematical abstraction — and an implementation of it as a virtual machine running on a physical computer. Implemented virtual machines have many of the characteristics of physical machines: they operate in real time at measurable speeds, can have causal interactions with their surroundings, and so on.)

For another example, many computers do not compute functions in the traditional sense of computability theory; they do not take inputs, process them and then

Page Break 391

produce outputs, but rather interact continuously with a changing environment, playing the role of a controller or an advisor.

Page 391

Computers contain and manipulate symbols. To a philosopher this is one of the most controversial claims and to a computer scientist one of the most obvious. This contrast needs to be accounted for. One way is to defend one or the other position; another is

to look for a difference in meaning. For example, one can say that computation has access only to the forms of these symbols, which are therefore mere patterns, and have no meaning to the computer. This has the makings of an argument against the "computationalist" approach to describing the mind, so has generated fierce opposition.
Page 391
One approach to describing a computer is as something whose physical or causal states mirror the state-transitions of a suitable abstract machine, defined mathematically. This has the attraction of allowing any suitable physical substrate, so that an abstract machine can be "implemented" in many ways. Many elaborations on this theme have been given, but I find them all unsatisfactory, for a number of reasons.
Page 391
First, this kind of characterisation is too restrictive. We don't have a single adequate abstract account of "computer," so any such account will miss some things: there will be computers which don't fit any particular abstract characterisation. For example, very few electronic computers have the state-transition structure of a Turing machine. Of course one can add extended abstract descriptions (multitape machines, branching automata, etc.), but clearly this process will never properly capture the notion of being a computer, just as no jeweller's catalog will capture the notion of "jewellery." One answer here is to refer to Church's thesis and the fact that there are classes of universal machines. But this confuses one of the distinctions mentioned earlier. There is no universal machine for computers, only for computability. That a Turing machine can compute any computable function does not mean that it can serve as an adequate model for every computational process.

> Selmer Bringsjord disagreed, objecting to the "jeweller's catalog" metaphor.
>
> [Bringsjord:] A jewellery catalog shows pictures of jewellry, and I agree that it's therefore implausible that such a catalog can serve as a *definition* of jewellery (though I suppose an *ostensive* definition ought at least to be considered in our deliberations). But the analog of the catalog in my [definition of computer], viz., the infinite collection of mathematical machines over which we have excellent command (Turing machines, Register machines, neural nets (with rational co-efficients),

Page Break 392
cellular automata, abaci, etc., ad infinitum), is *not* offered as the definition. The "catalog," in this case, is designed to convey the essence of computerhood; an account of how that essence figures in the definition in question is the definition!
Second. On the other hand, this kind of characterisation is also not restrictive enough. It allows too many physical systems to be considered as computers. For example, Putnam has observed that any physical object — say, a rock — can be regarded as an implementation of any finite-state computational process. While the details of Putnam's argument have been criticised and its scope shown to be limited, it still carries some force. For example, one objection is that the argument only works for a finite-state machine and fails as soon as there are inputs and outputs. But consider a limited part of a single computation, say the process of multiplying 27 by 3. This is describable by a finite-state machine, so Putnam's construction can now be given: but surely it is ridiculous to be able to interpret a rock as performing any computation, even one this simple and particular.

> [Bringsjord:] I have dealt with this problem in painstaking and (by my lights) efficacious detail in (Bringsjord, 1995). The trick is that my definition of computerhood — (D3), in the paper — contains a possibility operator that functions as a parameter that can be adjusted depending upon how strict one wants to be on the Putnam issue. The basic idea is that on some senses of "possible," a group of suitably configured pieces of granite becomes a computer, while on other senses of this term such a composite object is *not* a computer.

Finally, this kind of account is intuitively backwards. We don't recognize something as being a computation because it is describable as a certain kind of machine: rather, we recognise some machines as performing computations. The concept of being a computer seems primary to that of Turing machine or finite-state machine or any other particular kind of machine. In the early part of this century, people were called "computers," and they earned a living by performing computations, often using specialised algorithms: the entire vocabulary was being used in more or less its modern sense before Turing was born.

> [Bringsjord:] This is very weak…. People had a notion of algorithm (computation, computer) before Turing (and relevant others). But it doesn't follow from this that a *definition* of computerhood (and related concepts) can't make reference to mathematical objects only discovered in the 20th century.]

Page Break 393

3. An alternative account

Page 393
Here I will give a sketch of an alternative account of what it is to be a computer.
Page 393
Leaving open the question of whether a nervous system is a computer, artificial computers have two parts: a memory and a processor. Turing machines model this in the familiar way. The "state-transition" way of describing computation focusses on the processor, relegating the memory to an aspect of the state. Suppose however we focus instead on the memory.
Page 393
A computer's memory contains patterns (including the "blank pattern") which are stable but labile, and it has the rather special property that changes to the patterns are under the control of other patterns: that is, some of them describe changes to be made to others; and when they do, the memory changes those patterns in the way described by the first ones. One kind of change, for example, is that a pattern shall be created which denotes some property of another pattern, such as the number of symbols in it, or the number of prime factors its denotation has when it is considered to be a numeral. In this account, then, the central notions are those of patterns, changes to patterns, syntactic properties and denotations of patterns. A computer is a machine which is so constructed that patterns can be put in it, and when they are, the changes they describe will in fact occur to them. Notice that this is a perfectly accurate description of a Von Neumann machine.
Page 393
If it were paper, it would be "magic paper" on which writing might spontaneously change, or new writing appear. If some of the written text is about the writing itself, the text will rearrange itself so that the assertions made about it become true. Write "Fish are lackiing in insight," then write "Change 'ii' to 'i'". The first sentence would become correct and the second sentence vanish.
Page 393

Different kinds of computer can be defined abstractly in terms of the formal syntax of their pattern language, the kinds of syntactic change they can perform, and the semantic theory which relates the former to the latter. I have been careful to avoid calling these patterns "symbols," but they have a syntax and some have denotations. Moreover, if we put such a computer in a 'grounding' situation, some of the patterns can become causally related to aspects of the physical environment in systematic ways. For example, we might imagine a memory machine attached to a blood-pressure monitoring machine in such a way that the patterns are numerals which denote the pressure, ordered in the memory so as to denote the temporal history. Other symbols might specify sequences of syntactic changes which produce new numerals denoting the rates of change of blood pressure, and so on. Notice that it is not particularly strange to refer to the denotations of the symbols.

Page Break 394

> (Istvan Berkeley objected to "pattern": [Berkeley:] It seems to me that the term "patterns" is horrifically imprecise, given the importance that the term plays in the thesis — after all, wall paper can be said to have patterns!
>
> He suggests that we substitute "representations," understood in a Dretskian fashion, i.e., an item which indicates how things stand with respect to some object, condition or magnitude.
>
> I use "pattern" more or less as a synonym for "symbol," but attempting to indicate that they might have complex and non-language-like syntax (what computer science calls "data structures"), and that there is no presupposition that these symbols are meaningful. Particular computers may be configured so that some of the data structures denote or represent, but a computer programmed to merely play around with meaningless empty lists is still a computer, for all that.)

"Magic paper" has some potential for being a more rewarding way of characterising computation, for several reasons.

Page 394

First, it concedes the fact that computers manipulate formal patterns according to their form, without thereby also conceding that a full account of the computational process cannot or must not refer to the denotations of the symbols.

Page 394

Second, it abstracts completely from any particular implementation structure. It is a higher-level description, referring only to the syntax of the expressions in memory, not the engineering minutiae.

Page 394

Third, this account makes symbols — or, patterns, at least — a central issue in computation, which they surely are. This is what people have always meant by the term, ever since a time when to be a computer was a matter of professional academic pride. That is why machines like EDSAC and JOHNNIAC were called computers: they were the first machines that could clearly perform symbolic computations.

Page 394

Fourth, it emphasises that being a computer involves more than moving appropriately between states. Many state-transition machines are not computers not because they fail to conform to a finite-state machine, but because they do not encode symbols: nothing in them has a syntax which refers to their own state-transitions.

Page 394

This is worth looking at more carefully, since in a sense this condition is trivially satisfied. In Putnam's spirit, we can define each state as being a 'pattern' which denotes the transition which immediately follows that state: the pattern-language has a trivial empty syntax, consisting only of a set of discrete symbols.

Page Break 395

Any finite-state system is then a "computer" in our sense; but note that the "memory" of such a computer has a recognisably trivial syntax, no matter how many states the machine may have. No matter how complex the machine, it is always trivial as a computer unless it has a nontrivial memory language, so that it becomes useful to describe it as containing patterns.

Page 395

Here ends the target article.

Page 395

This gave rise to several issues in the subsequent discussion, as well as those in the comments already inserted.

4. Other architectures

Page 395

Some gave highly non-Von-Neumannesque examples of computation which seem to escape any such attempt at a definition. While the "magic paper" account was intended to abstract as far as possible from low-level minutiae of how things are implemented, even talk of "patterns" may be too much for some. Rob Stufflebeam suggested that we simply define a computer as broadly as possible:

> [Stufflebeam:] Something is a computer just in case it can be described as satisfying or implementing some function F. Since practically everything in the universe can be described as implementing some function or other, it's hard to imagine a more general — and unambigious — definition.

Selmer Bringsjord replied (correctly, in my view) that this was too general, and that many real computers do not in fact compute functions, but perform such tasks as real-time control of ongoing processes.

Page 395

McKee also offered a very general definition: something is a computer just when it can be programmed. This seems to concede Searle's view of original intentionality rather directly, until one realizes that "programmed" here means only that information from somewhere else influences the computer's behaviour:

> [George McKee:] All that is needed is compartmentalization between programmer and target systems. One can imagine a purely biochemical computer that operates via enzyme-linked metabolic networks. The elements of these networks exist in the same (unpartitioned) reaction cell, operating by transforming one chemical species into another. The programming module could operate over one set of species (consider "aminergic") and the target

system could operate over a different set (consider "cholinergic") as long as there were a defined (set of) communication channel(s) between them.

Page Break 396

Another example might be the biochemical computers that can use DNA polymerase reactions ("PCR") to achieve parallelism dozens of orders of magnitude greater than has been achieved in silicon-based computers. These systems have significant problems with i/o and issues of universality, but instances have actually been built, and used to solve small instances of NP-complete problems.

This means that in order to have a notion of computation that is rigorously consistent with all that we know about the physical world, it is necessary to recast most of the common notions about finite state machines and truth-functional operators in terms of continuum mathematics and probability distributions.

McKee finds this recasting means that "it's an exciting time to be doing research." I confess I find this prospect depressing, but readers must make their own judgements.

5. Are brains computers?

Page 396

Valerie Hartcastle and Istvan Berkely both regard this as a matter of definition:

> [Hartcastle:] I take it that part of what is driving this discussion is the question whether our brains could be a computer. It would be unfortunate if our definition of what a computer is ruled it out of hand, contrary to any well-developed and well-informed intuitions regarding this matter.
>
> [Berkeley:] It seems to me to be a requirement that brains should count as being computational in at least some respects. This is because, as Hayes' noted (Sec. 3.9), machines like IBM XTs are called computers due to their similarities to human computers. It would seem to me that an account of computation which failed to capture this intuition would simply be defective.

However, they disagree as to whether this posed a problem for the magic paper account:

> [Hartcastle:] Hayes claims that computers have to operate over patterns that behave as syntactic formuli. Whether connectionist machines have syntax and can be described as manipulating patterns to produce output is a well-known dispute in philosophy and cognitive science. I don't want to answer that question here; I don't know the

Page Break 397

answer. I do though want to flag this as an area of concern. Hayes appears to have already assumed the answer is yes in the definition. Worse, if brains work at all like the way Walter Freeman thinks, for example, then Hayes's definition is really in trouble (or brains just aren't computers). If our cognitive currency can be described as limit cycles in phase space whose patterns are generated and maintained by the underlying activation of neurons (but not by any particular pattern of individual neuronal activity), then it is difficult, if not impossible, to describe what is happening as operations over syntax.

This isn't obvious. If one looks closely enough at the way a computer works, the symbols in the memory will not be instantiated by any particular pattern of electrical charge, especially if the machine does clever things like memory paging and online file management. Since a computational simulation of a computer is itself a computer (much as my Powerbook consists mostly of software), it could well be that the computer described by Freeman is itself rather a simple device serving as an interpreter for a more complex machine whose "cognitive currency" consists of symbols which are quite intricate dynamic patterns in the simpler machine's memory.

Page 397

However, all of this is conjecture, and I concede to Hartcastle that if something like Freeman's model is accepted as the proper account of brain functioning, then it becomes a nontrivial task to show how the brain can be a syntactic computer. It seems clear that the human CNS is capable of producing and processing syntax, and of encoding information that is externally expressed in syntax, all of which strongly suggest that it might work by storing and manipulating syntax; but this is not a conclusive demonstration.

> [Berkeley:] Hayes position makes it quite clear that PDP models should count as computational too. Firstly, most connectionist systems are instantiated as virtual machines upon Von Neumann architectures, and as a consequence would count as computational. Secondly, what little evidence that exists with respect to the internal structures of connectionist systems, seems to indicate that such systems contain representations (in Dretske's sense) and may even, in certain circumstances be described as having operations which manipulate those representations. Indeed, in some cases I would go as far as claiming that connectionist systems contain "rules." (Berkeley et al. 1995)

Page Break 398

6. Possible and actual computers

Page 398

There was some discussion about the role of modal or counterfactuals in a proper definition. Istvan Berkeley drew attention to the case of a computer which comes off the assembly line but is never switched on:

> [Berkeley:] Under such conditions, the notion of "changes in pattern" would not be relevant as there would never be any patterns present to change! Such a device would be a pretty much prototypical example of a "computer," even though it would never perform a computation. What I think this shows is that the notions of "computation" and "being a computer" are to some degree independent of one another. Perhaps the best way of describing the relationship between the two would be to say that something X is a computer just in case it is possible that X may perform a computation (understood in Hayes' sense).

(This is what I meant to suggest by saying that patterns can be put into it.) Bringsjord (1995), in a previous paper on this topic, develops this point in more detail. As he emphasises there, it is important to choose the right modality. Logical possibility would be too weak, for example. Physical possibility seems about right; but given this, Bringsjord claims that we would not need to add Berkeley's qualifier "in Hayes' sense," but simply understand "perform a computation" in the traditional sense, referring to Turing machines.

Page 398

This does not seem to avoid the Putnam construction, however, since there seems to be nothing physically impossible in interpreting successive moments in a pebble's history as states of a finite-state machine, no matter how ridiculous this may seem. Valerie Hartcastle also had worries about how to choose the right kind of possibility:

> [Hartcastle:] I worry though that (Bringsjord's) definition isn't too broad as well. That is, I don't think that it would block the very examples that Searle uses (at least in his public talks on this matter) of a set of molecules instantiating Wordstar. (Some of this might turn on how we define the notions of agent or observer.) Okay, so we move to the most restrictive notion of possible: humanly possible. I see this as too strong, for exactly the sort of reasons McGinn gives for us not being able to understand consciousness: we are too stupid. I would think that we would want to count as a computer some things that we in fact can't manage to analyze (because of some inherent physical limitations: we are too stupid, or too slow, or we don't live long

Page Break 399

enough, or whatever). So, I am looking for a notion of possibility somewhere between physically possible and humanly possible. The case of the machine which is never switched on seems to arise for any definition of computer, rather like an automobile engine which never runs, or a hat which is never worn. I confess that this seems to me to be a rather unrewarding issue. My inclination would be to seek to define the notion of "working computer," and let philosophical logicians quarrel about such cases.

7. Objective computerhood?

Page 399

Several authors urged that whether or not something is a computer is not a matter of objective fact, but depends on what attitude one brings to the question.

> [Valerie Hartcastle:] We define computations and computers relative to some specific set of explanatory concerns. That is, it is largely a pragmatic decision whether we consider the digestive system a computational device, as is whether we consider my Mac a computational device. How we talk about things depends upon what about those things we are trying to explain.
>
> Searle would counter that this view makes computation a purely artificial kind; that is, there is nothing out there that makes anything computational. This means that computers and computations are a different breed of concept than our real scientific notions, like force or mass or species, which presumably do exist apart from our pragmatic concerns.
>
> My response to Searle is that he misunderstands science. Scientific theories are not formed in a vaccuum. They all are produced in a particular context and are designed to answer certain questions (with a particular contrast class) and not others. Furthermore, they are abstract idealizations of some phenomena we have observed (or would like to observe). Consequently, physicists use the term "force" or "mass" in particular contexts when answering particular questions and these terms get their meanings from those particular usages. Similarly, cognitive scientists (say) use the term "computer" or "computation" when referring to certain abstractions embedded in a particular sort of explanation, given to a particular sort of audience. In sum, what Searle complains about as the problem with the notion of "computer" is a "problem" for all scientific notions. (And hence, not really a problem at all.)

Page Break 400

Istvan Berkeley suggested that that the notions of "computation" and "computer" are inherently ambiguous.

> [Berkeley:] I think that this ambiguity might explain why there are conflicting intuitions about the suitability of definitions, and also explain why Searle believes that he can make the kind of claims he does.
>
> I am pretty certain that the term "computer refers to a number of distinct classes of objects. Perhaps it would be useful to attempt to find a set of definitions for "computation" which served to capture different aspects of the entire class of computational entities. I certainly do not feel up to this task alone, but what I will try and do here is offer two versions of the notion of computation, which seem to lie at the extreme range of the term.
>
> The first notion of computation, which I will call computationT, short for Technical computation, is the most restrictive notion. Entities which have the properties of a Turing machine or perhaps a von Neumann machine would fall within the scope of this notion. An entity is computationalT if it engages in the rule-governed manipulation of complexes of structured symbols (this is vague, I know — please fill in your own favorite list of properties, as appropriate). Many connectionist systems might fail to be computational in this sense (assuming that the claims made in the standard connectionist literature are correct, something which I am not too convinced of). Similarly, it is far from clear whether biological cognition would be an instance of computationT (Pylyshyn would probably say "yes," but others would say "no"). On this restricted notion of computation though, it is clear that Searle and Putnam's Wordstar-instantiating walls and computational stones would be ruled out.
>
> There is another notion of computation which is also common though. This I want to call computationC, short for common sense computation. This notion of computation is about as broad as is possible. Roughly, something is computationalC if it is something which can be done by a computer. So, to cite a ridiculous example, playing a Bach fugue would count as being computationalC, as a machine with a sound card, CD drive etc., can play a Bach fugue.

> Pretty clearly this notion of computation is way too broad to be of much use for anything serious, as it makes many too many things ComputationalC.

Page Break 401

Well, maybe not. Attacks on the "computationalist hypothesis" by Searle and others sometimes confuse these senses. The philosophical literature has taken to using the term "computational" as an adjective on kinds of machine, but confuses it somehow with "computable," which is a mathematical property of functions on the integers. Computability is connected with Berkeley's computationT, but arguments about what a computer can or cannot do seem to be inevitably referring to computationC. (My Powerbook can receive a Fax transmission, an ability not mentioned anywhere in the theory of computation; fax-reception is clearly computational, but — since it has nothing particularly to do with functions on the positive integers — not computable.)

Page 401

Until we can say what a computer is, computationalC is meaningless, unfortunately. Which brings us back full circle to where we began. Also, it is not obvious that computationT rules out the Searle-Putnam examples. The final point of discussion is whether any of these proposed definitions manages to avoid Searle and Putnam.

8. Reducible to absurdity?

Page 401

Part of the motivation for finding a better account of being a computer, like the 'magic-paper' account I sketch, is precisely to capture the intuition that the digestive system, Searle's office wall, etc., *are not* computers. (Or if they are, they are extremely trivial computers, rather as a rock is a computer with one symbol which means "do nothing.") I take this as simply obvious: my digestive system doesn't, in fact, do word processing or compute tables of prime numbers. If our existing accounts of computer and computation predict that it does, or even can be seen that way, there is something wrong with them. Rob Shufflebeam agrees:

> [Stufflebeam:] Physical devices/systems don't suddenly become symbolic-digital computers because their behavior has a symbolic interpretation. The difference between as-if symbolic computation and real symbolic computation is a difference that makes a difference.
>
> [Hartcastle:] Though I am sympathetic to Hayes' general approach (computations are things that computers do, so we need to define what a computer is first), I worry that the answer provided does not adequately counter the critic's claims.
>
> First, regarding the specific suggestion that to be a computer requires a certain level of complexity: this does not seem to answer Searle's real challenge, viz., that his digestive system does everything that Hayes claims a computer does, yet we don't think of digestive systems as being computers.

Page Break 402

To avoid triviality, however, a magic-paper computer has to be not just complex in the sense of having a large number of physical states, but also these states must exhibit tokens of a language with a productive syntax which can be interpreted as referring to the state-transitions which are the causal consequences of the state. A machine becomes recognisable as a computer precisely when this data-structure language is more complex than the trivial language obtained by assigning a unique token to each state, so that it can be accurately described as storing and processing information. This is a much more demanding criterion for any proposed Putnamor Searle-type construction.

Page 402

However, this is only a sketch, and details are not yet worked out; so I will join with Valerie Hartcastle in her last word:

> [Hartcastle:] I raise more questions than I answer, and for that I also apologize.

9. Coda: What went wrong?

Page 402

This particular e-mail discussion has not been a very successful event, in spite of the quality of the contributions. It attracted only a small number of participants and never seemed to take on a life of its own, eventually fading away to silence. It seems worth trying to analyse some of the reasons for this.

Page 402

First, if there is any blame, it must of course rest with me. At a critical point I was unable to access the e-mail network regularly, and this is when my intervention was probably most needed. Evidently there is more to making an electronic discussion work than simply posting some controversial ideas and hoping that they will stimulate a lively discussion. In retrospect it is clear that I should have been quicker to intervene directly with questions, comments and general promptings (and to have more actively sought contributions from those who had earlier expressed an interest in the topic but never, in fact, became actively involved.)

Page 402

Rather as at a successful party, the host cannot simply join in the festivities, but must be constantly scanning for potential problems or lapses in the *joi de vivre* and ready to intervene deftly in order to keep things going. The really good host, like the really good moderator, does this so cleverly that it appears effortless; this self-effacing skill is the most successful when it is almost invisible, which misleads the naive observer into thinking that there is nothing to it. But to organise a successful electronic meeting demands at least as much skill and energy as organising a good party. (One particular problem: I found it difficult to take part in the ongoing discussion with my usual zeal, as such a degree of partiality and enthusiasm seemed to be at odds with the overall sympathetic objectivity expected from a moderator. It is difficult to be simultaneously a player and a referee.)

Page Break 403

Page 403

However, there were several more objective reasons why this discussion was probably doomed to fail. A very successful electronic discussion organized by Stevan Harnad on almost the identical topic had recently been published. Several of the contributors to that discussion had gone on to develop their particular views in more conventional papers, or were now even working on books. Several of the contributors to this discussion therefore began by referring the reader to these publications. While reasonable under the circumstances, this is probably not a good way to begin a lively conversation. E-mail communication lies somewhere between writing and talking, and seems to be best when it can bring together some of the the precision of written text with the

liveliness and spontaneity of a face-to-face conversation. If we begin by handing one another reprints, it takes an unusual degree of determination and self-control to read them all before saying anything.
Page 403

There seems to be a natural progression revealed here. A topic appears which sparks widespread interest and on which not much has been written. It may arise from a notorious academic assault on some established position (as this did from Searle's claim to have reduced computationalism to triviality), or by a publication which gives academic blessing to a topic which has been déclassé for many years (as Dennett and Chalmers recently did to consciousness): but whatever, many people have views and enjoy a debate; and at this stage, debating is indeed a useful mechanism. It allows the rapid exchange of many starting ideas and counterar-guments, exposes many initial misunderstandings, and allows people to sort out the major "positions" which are sensible and coherent. This is when electronic discussions are uniquely useful and productive.
Page 403

But then there comes a more mature stage, when the various positions need time to develop in a more careful and scholarly fashion. This is slower, produces longer texts and tends to display firmer opinions. While electronic communication will always be uniquely useful between individuals, an e-mail forum does not seem to be so natural at this stage. Many of the people who might have made it interesting now find it a distraction. I think that this question of how best to characterise computation was already well into this second stage even before we began.
Page 403

Editor: *Patrick J. Hayes*

University of West Florida
Page 403

Authors:
Istvan Berkeley

Page Break 404
Page 404

Selmer Bringsjord
Valerie Hartcastle
George McKee
Rob Stufflebeam

REFERENCES

Page 404
Bringsjord, S., 1995. "Computation, Among Other Things, Is Beneath Us," *Minds and Machines*, 4, 469–488.
Page 404
Berkeley et al., 1995. "Density plots of hidden value unit activations reveal interpretable bands," *Connection Science*, 7.2, 167–186.
Page 404
E-Mail Addresses: Patrick J. Hayes <phayes@ai.uwf.edu> Istvan Berkeley <istvan@psych.ualberta.ca> Selmer Bringsjord <brings@rpi.edu> Valerie Hartcastle <valerie@vt.edu> George McKee <mckee@starbase.neosoft.com> Rob Stufflebeam <rob@twinearth.wustl.edu>

Page Break 405