A student writes:
I'm still troubled regarding Moor's article. In particular, his account for computer vs computer program. I'll refer back to the LISP example. To run a lisp program, we first need a LISP interpreter.

As I mentioned in lecture on Monday, that's not necessarily the case; Lisp machines used Lisp as their machine language, so didn't need interpreters. See http://en.wikipedia.org/wiki/Lisp_machine But in general you're right (though, of course, sometimes it's a compiler that's needed, rather than an interpreter—the difference is the same as the difference between a complete translation of a text from one language to another ("compilation") and a sentence-by-sentence translation ("interpretation")).

Eventually at some point we go through interpreters until we reach a point of Physicality and no longer need something which can 'understand' our program. Once we hit that point, we are said to be at the computer.

I'd say it's physical all the way down. But your main claim, I take it, is that there has to be a stopping point, and that the stopping point is the computer itself. Ray Turner, in his last lecture, called this "multicircular" interpretation: Language L1 is interpreted into L2, which is interpreted into L3, …, which is interpreted into Ln, which is, ultimately, at last, interpreted by the computing machine M. And, presumably, M is not itself a language, but a device that does something with the language Ln that it "understands"—i.e., that it knows how to do something with.

(I've made a similar point in my objections to John Searle's "Chinese Room Argument"—which we'll look at later this semester—only I claim that when humans (or computers) understand natural language, that bottom level is also a language—what the philosopher Jerry Fodor calls the "language of thought" or "mentalese". It can be thought of as the language of neuron firings. And that language doesn't require another language to understand it; we understand it just by having gotten "used to" it. For more details on this, see:
Rapaport, William J. (1995), "Understanding Understanding: Syntactic Semantics and Computational Cognition", in James E. Tomberlin (ed.), *AI, Connectionism, and Philosophical Psychology*, **Philosophical Perspectives** Vol. 9 (Atascadero, CA: Ridgeview): 49-88.

http://www.cse.buffalo.edu/~rapaport/Papers/rapaport95-uu.pdf (By the way, this is the paper of mine that I told you about; it was reprinted in another anthology without my knowledge or approval because I no longer own the copyright—not that I minded!)

This to me is problematic, and I don't particularly like the results. It would seem the best case scenario is that there is only one large computer(the universe) and we are placed into an uncomfortable predicament with our language.

Now I'm puzzled: There are 2 claims here: "There is only one computer (the universe)" and "We are placed in an uncomfortable predicament with our language". Which of these (or is it both) is "problematic"? And why? The first may be odd, but Seth Lloyd and Stephen Wolfram wouldn't be upset by it. And the second is unclear: What is the "uncomfortable predicament"?

The worst case scenario is that we have an infinite regress and there are no computers.

Agreed: We need to avoid the infinite regress. Although one possibility that may not avoid it is to say that, while we use one language (L2) to understand another (L1), we don't have to understand L2 semantically—we only have to be able to use it syntactically. (That's sort of the line I argue in my 1995 paper.) But even if we have an infinite regress, at each step we could have a computer, albeit one that we can't understand except moving to the next level.

Suppose we have a LISP program running on your off the shelf x86 platform. At what point does the interpretation start being computed by the computer?

Let's be careful here. At first, I thought you were asking when the **program** starts being computed. But you seem to be asking instead when the **interpretation** of the program starts being computed, i.e., when the machine-language implementation of the Lisp program starts being computed. Am I right in my interpretation of what you wrote?

CSE Department 3/22/10 11:24 AM
Formatted: Font:Bold

CSE Department 3/22/10 11:25 AM
Formatted: Font:Bold

Supposedly it's once the instructions reach the CPU and at that point no longer need an interpreter. However I will argue that even the CPU(and any other device for that matter) is in fact in need of an interpreter.

I’m not sure what you mean here: An interpreter takes one **language** and translates it into another **language**. But the CPU is a machine, not a language. So how can it be “interpreted”?

CSE Department 3/22/10 11:26 AM
Formatted: Font:Bold

CSE Department 3/22/10 11:26 AM
Formatted: Font:Bold

Namely, if we embrace the modern physics perspective, the CPU is reliant upon a lower layer of say molecules. This molecular layer decomposes into various elements, atoms, protons/neutrons/electrons, quarks, boson's mesons, etc. In order for the CPU to execute, there must be a lower laying executing the CPU.

Here, you’re talking about what philosophers of science call “reduction”, rather than translation or interpretation. The CPU is, of course, constructed from (“reducible to”) quarks, etc., and, in some sense, it’s the quarks, etc., that are “really” doing the computation. I’m willing to go along with you on this, but keep in mind the different “levels” or “stances” that we talked about in connection with Moor and Dennett: Just because you **can** describe a machine at the level of quarks doesn’t mean that that’s the most illuminating level to describe certain aspects of its behavior. Two different CPUs that behave identically will be made of different quarks, yet will be said to “do the same thing” at a higher, more abstract, level of description.

CSE Department 3/22/10 11:29 AM
Formatted: Font:Bold

Just as there needs to be a lower layer to implement any high level programming language. Now let us suppose there is some truly fundamental particle. It follows that in reality the universe then is one big computer composed of said particle. All of our concrete particulars in said universe are just interpreted programs. In reality there would be only one 'computer', and when we speak of our x86 computers we are really speaking about programs. This is the less harmful, but nonetheless uncomfortable scenario.

Again, why is this “uncomfortable”? It seems to me that it could be true. If it is, it would still be useful and/or necessary to talk about the behaviors in the universe at a higher, more abstract level. This, by the way, is the fundamental insight of “functionalism” in the philosophy of mind: Minds are, at one level, implemented in neurons and neuronal behaviors; but the psychological level of description of their behavior is sometimes more useful.

The more harmful scenario runs as follows. Inductively it appears to me that with any modern scientific atomism comes an infinitely recurring sequence of smaller and smaller pieces. Given infinite time for discovery, we will commit our selves to an infinite number of 'basic' particles. If you accept my sloppy inductive proof, it would seem there is no end to the interpretive process.

I’m glad you admit it’s “sloppy”☺ It certainly **seems** as if there might be no end to the kinds of “elementary” particles. But there might be. But let’s suppose there isn’t; then what? You go on to say:

CSE Department 3/22/10 11:33 AM
Formatted: Font:Lucida Grande, 13 pt, Font color: Black

CSE Department 3/22/10 11:33 AM
Formatted: Font:Bold

As a result, we would commit ourselves to the existence of only computer programs.

Here, you’ve lost me; I don’t see how this follows. Unless you are speaking somewhat metaphorically and mean something like “we are committed to the existence only of intermediate-level entities”?

Further more, we would commit our selves to the non-existence of computers.

Do you mean that we would commit ourselves to the non-existence of the bottom level? Yes, that would follow from your assumption.

As a result, we have only computer programs without a computer to run them on. How can it be the case that this is possible? If you haven't figured out by now, I'm arguing it isn't.

Well, would the notion of a virtual machine help? Each intermediate level would be a “language” or “program” running on a virtual machine at the next level down. Either this bottoms out in some “final” machine that’s really running things, or it goes on forever; but if it goes on forever, maybe it could still “work”. (Now I’m getting vague; sorry!)

It may be argued that we need not hit the lower most layer of physicality to declare ourselves in the computer.

What do you mean by “declare ourselves **in** the computer”?

However, should this be the case, why not push our primitive higher up the ontology? It seems to me that any computer program must itself be physically represented in some form(baring the existence of a program in the mind and mind's being non-physical). There would appear to be no reasoning as to why it must be placed near the CPU/Registers/etc. I would argue that A CPU is a grouping of transistors, those transistors are really just a grouping of elements, those elements atoms... A computer program may just be a series of bits on a hard drive, those bits a magnetic orientation, that orientation an arrangement of electrons/protons and on and on. Or if you want to account for programs on paper, a program may exist as a series of symbols, those symbols then made of ink on paper, and on and on.

One possible objection I do see, is to maintain that some how programs exist in some sort of other worldly plane of existence. Really a program exists there, and the physical computers just model it. However, I find myself opposed to these sort of super natural and apriori states of affairs. Also, I see no reasoning why this ought to be adopted except for the ability to save the prior account of a computer/computer program. However, if that should be our only reason for introducing this new plane of existence, I should think we would be better off finding a different account for computer/computer program. It would seem that introducing such a realm provides us with no greater explanatory power.

Here, I think I lost your meaning.