UNIVERSITY OF TORONTO
FACULTY OF INFORMATION
Coach House Institute

Computer Science, AI, and Philosophy of Science

Remarks to be read at Panel #3 of the conference on
The "Stanford School" of Philosophy of Science
October 25–26, 2013

Brian Cantwell Smith

I am immensely sorry not to be able to join you all today, for at least three reasons.

First, I regret not being able to revisit a place where I spent sixteen wonderful years. To all my dear friends and erstwhile colleagues: please know how very much I wish I could be with you today.

Second, it would have been wonderful to have a chance to reflect on my collaboration, and that of other computer scientists and AI researchers at Xerox PARC, SRI, and Stanford, with the philosophers at Stanford (and at SRI: hi David!). Those early CSLI days were the most intense and productive intellectual years of my life. Deeply treasured, and still sorely missed.

Third, I regret not being able to join you because this would have been an ideal context in which to probe a question about which I have long wondered: why it is that computer science and artificial intelligence, over their 50 year history, have had such close associations with logic and philosophy of mind, and to a lesser extent with philosophy of language, but so little connection with philosophy of science.

That pattern certainly held of me. I collaborated closely with (especially) Jon Barwise and John Perry, and with Fred Dretske, and to a certain extent with various CSLI linguists (primarily Geoff Nunberg, Joan Bresnan, Stanley Peters, and Ivan Sag). But connections with John Dupré, Pat Suppes, Nancy Cartwright, and other core members of the "Stanford School" were far more minimal.

It is not a fact I condone. In retrospect I wish the engagement had been much stronger. But I think the *reasons* for the asymmetric history are telling—and perhaps pertinent to this conference's agenda.

A hint can be seen in vocabulary. Note that the technical terminology of computer science stems from the *rationalist*, not *empirical*, tradition. Consider the work-a-day terms used by both practical and theoretical computer scientists: *language*, *symbols*, *reference*, *syntax*, *semantics*, *interpretation*, *meaning*, *data*, *representation*, etc. (along with such mathematical notions as *number*, *function*, *computability*, etc.). These notions are not at all like *mass*, *energy*, *nuclear-magnetic resonance*, or *benzene rings*—or even *adaptability* or *fitness landscapes*. More generally, they are not obviously concepts or phenomena

 brian.cantwell.smith@utoronto.ca

located in the causal realm (more carefully: not obviously *subsumable under causal explanation*). Issues of meaning and interpretation—and to some extent normative issues of worth and value—are much more in the driver's seat than cause, effect, or anything related to bumping and shoving.

Or so, anyway, the technical vocabulary would suggest. And so, too, I myself thought, when I was there. Not that materiality and concreteness were unimportant. I was hugely influenced by issues of essential indexicality, by differences between meaning and interpretation, by the roles of content and character, by what it meant to be "situated," etc. And think about Fred Dretske's wrestling with how to "put information to work," and John Perry's and David Israel's comments about how, even if you take information to be fundamentally semantical (counter-factual correlation or whatever), the *form* that information takes remains critical. And so on. Perhaps it was because these causal issues remained philosophically recognizable, but intentional issues given pride of place, that it was logic and philosophy of mind & language to which we were all drawn.

Put it this way: in spite of our eponymous name, it basically never occurred to me that philosophy of science might be computer science's philosophy.

That terminological issue was confounded, moreover, by a related one that plagued CSLI from the beginning: the fact those of us from different fields used the same terms to mean different things. I remember drawing up a matrix, for early Symbolic Systems students, crossing fields, on one axis (philosophy, psychology, mathematics, biology, computer science, linguistics, artificial intelligence, etc.), with terms, on the other (*function*, *concept*, *intension*, *purpose*, *meaning*, etc.)—and in each resulting entry taking a stab at explaining how that term was used in that discipline. I also remember suggesting, at an early CSLI tea, that we institute a "technical term library," so that if anyone wanted to use a word of English for technical purposes, they had to sign it out—and say when they were going to return it. Yet the problem persisted. It wasn't until our second year, if I recall correctly, that we realized that by 'theoretical' the Stanford people understood "not experimental," whereas the PARC and SRI contingents thought we had always meant "not *applied*." And it was even later, in the midst of a technical design project, that Barwise and I were brought up short by discovering that logicians and computer scientists meant different things by "binding a variable."

Yet another more systematic terminological issue confounded relations between computer science and philosophy. It turns out, in ways I understand much better now than I did in the 1980s, that all the technical terms I mentioned above—*meaning*, *semantics*, *interpretation*, *reference*, etc. (even ontology)—have been *redefined* in computer science to name *local, immediate*, *causal* relations. The "semantics" of a program, for example, names nothing having to do with any relation to the outside world or program "task domain," but to that between a static program and the *causally-individuated local behaviour* that arises from running it.[1] A variable labeled 'current-floor' in an elevator

[1]This is true of both operational and denotational semantics, the difference between which is absolutely

control program is taken to *denote a memory register*, not, as one might expect, to denote the floor the elevator is currently at. And so on.

The long and short of it is that, the more I was steeped in "real" semantics and intentionality by the Stanford philosophers, the less I grew able to communicate with my fellow computer scientists. Since they meant different things by the words I was using, their understanding inevitably parted company with my intent. Moreover, since philosophy is not something you ever get over, I *still* can't talk to them. Just recently I gave a talk at the computer science department, here in Toronto, and was resoundingly scolded, in the Q&A session, with; "Don't you realize that you are hopelessly out of date? That's not what semantics and reference *are*, any more. Just get over it!" (Linear logic, a computational favourite, is another great example of the terminological—and conceptual—disconnect. Linear logic does not license the inference from P to P∧P. "Of course that's not valid," computer scientists say. "Suppose P were a dollar. From one dollar you can't just get two dollars." I trust it is clear that something has gone seriously awry.)

In sum, although (i) the *terminology* of computing is almost wholly intentional, and (ii) the real-world *phenomenon of computing* is genuinely intentional, in my view, (iii) the theoretical concepts and apparatus of computer science—perhaps under the impression that to be a science requires pledging a priori allegiance to causal explanation—treats its subject matter as a bluntly causal, bumping & shoving phenomenon.

Had we been able to draw the logicians, philosophers of language and mind, computer scientists, and AI researchers together into a conversation with the philosophers of science, we might have been able to sort this out—pursue questions that even to date are rarely asked. For example: does computer science count as a natural science? Or does it, too, need naturalizing? Or: why is it that most computer scientists, but relatively few philosophers (in my experience), consider the relative computability/complexity results to be more significant than the absolute ones? And so on.

There are myriad more reasons to wish for thicker connection between philosophy of science, on the one hand, and computer science and perhaps especially AI, on the other. Here's another: if AI is about *human reasoning*, why does it focus so strongly on *valid logical inference*, rather than on what one might have thought would be much more germane: treatments of evidence, theories of justification, contexts of discovery, and so on? It would not go astray, I should have thought, to argue that philosophy of science might have more to contribute to our understanding of day-to-day rationality than formal models of truth preservation and logical entailment.

Ontology is another example: things disordered, worlds dappled, what-all socially constructed, and the like. During the 1980s (along with at least Barwise), I felt that the most serious issues facing computer science and AI were *semantical*. I now believe that

---

not equivalent to—not even *related* to—the logical distinction between proof theory and model theory, or syntax and (logical) semantics.

the ontological challenges run even deeper. Being intentional, computer and AI systems must represent or in some other way deal intentionally with the world—which requires framing and making commitments about how the world is to be taken. But while I now use Dupré and Cartwright and Hacking in my teaching, I don't believe that any of us did, back then. It was only towards the end of the 1990s, by which time Barwise and I were both in Indiana, that, though our friendship remained strong, our theoretical work started to diverge, profoundly, over such ontological issues. I became more and more convinced that what AI and computer science needed was an ontological picture closer to that of Dupré and Cartwright and Hacking than to the mathematical worlds that Jon inhabited (except towards the very end his life).

A third issue of philosophical interest is the extent to which computer science is as much *engineering* as *science*. Rather that being permeated with the fundamental humility I associate with science—that if one's theory and the world part company, one should revise one's theory—computer science's reaction to such a discrepancy is often that we should *debug the world*. What does that do to truth, or even to naturalization? What is it to asses whether a theory of computation is *correct*?

For these and many other reasons, while it was an amazing, productive, and fecund time, it will forever be a regret of mine that, in the heyday of what we are here celebrating as the "Stanford School," there wasn't more engagement between the computer science and ai community and philosophy of science.

Huge thanks, though, to Rasmus and Lanier, who will perhaps not only honour to the history, but move to repair those omissions, and move us all productively forward. That is something that I, at least, will truly celebrate.

My very best to each and every one of you.

Brian Cantwell Smith
Director, Coach House Institute
Professor of Information, Philosophy, and Computer Science
Senior Fellow, Massey College
University of Toronto