How Much AI Does a Cognitive Science Major Need to Know?

John Batali
Department of Cognitive Science
University of California at San Diego
9600 Gilman Drive
La Jolla, CA 92093-0515
(619) 534-7308
batali@cogsci.ucsd.edu

Introduction

Artificial intelligence has given the world a set of programming tools and practices, a set of experiments and models, a set of experiences, and a set of intuitions about what intelligence, representation, and computation are about and are not about. What of this legacy is still relevant and crucial to a cognitive science undergraduate?

I should point out right away that the topic of this essay is what is often called "traditional" or "symbolic" or "good-old-fashioned" artificial intelligence. At least from the vantage point of La Jolla, such artificial intelligence has fallen on hard times of late — certainly it is out of style — and it is difficult to support the claim that artificial intelligence can provide scientifically viable models of human or animal cognition.

In this essay I first review our departments computation curriculum, discuss whether artificial intelligence could ever form the basis for theories of cognition, and present a set of specific artificial intelligence topics that I believe should be covered in the computation part of a cognitive science curriculum.

The Computation Curriculum of the UCSD CogSci Department

Our department's curriculum is organized around a three-way division of cognitive science into what we call "brain, behavior, and computation." Our students take a minimum of four courses in each of these areas. The "brain" courses cover neuroanatomy and neurophysiology, functional systems and cognitive neuroscience. The "behavior" courses cover experimental methodology (i.e., statistics), cognitive psychology, social psychology, anthropology, and linguistics. The "computation" courses cover introductory programming (in LISP), the theory of computation and formal systems, parallel distributed processing, and artificial intelligence. The same division of courses is used in our graduate foundations series, with the material being covered as a faster pace and in more depth.

As I mentioned, our introductory programming course uses LISP. This makes sense for a lot of reasons, the most importance of which is the ability to get to interesting issues quickly (like recursion, and structured objects, and interpreters) without spending a great deal of time dealing with syntax or the compiler. A few AI issues are introduced in this course, for example pattern-matching databases, and simple search. For the most part they are used to illustrate programming issues, not as especially relevant to cognition.

Our "theory of computation and formal systems" course is new (and hasn't actually been taught yet). The idea for this course is to cover formal logic for about a third of the time, and to divide the remaining among topics like: Turing machines and theory of computability; computer architecture; limitative results of formal systems (Gödel's and Tarski's theorems). This course does not involve programming assignments.

The required course in "artificial intelligence modeling" currently concentrates on: symbolic programming, search, structured knowledge representation systems, and computational linguistics. Our elective course in "advanced artificial intelligence" focuses mostly on logic and other applications of "declarative representation systems": unification, resolution, truth-maintenance systems, qualitative modeling, and version-space learning.

We have two courses in parallel distributed processing, one required and one elective. We also have elective computational courses in artificial life and language comprehension.

Students come to our artificial intelligence courses with a fair amount of apprehension. For one thing, some of our students are not primarily interested in the computational aspects of our department — given the three-way split of our courses, it is natural that some of the students prefer one or both of the other two. Still our undergraduates understand the importance of computation (or presumably they would be majoring in psychology or neuroscience). The AI courses, in particular, suffer from the fact that UCSD is known as a "neural networks" campus, and the enthusiasm for neural networks extends to most undergraduate students who have heard of it, even if their understanding is inchoate. Still, it seems that most of the students

in our AI courses find them interesting, and the programming assignments challenging and worthwhile.

Those students in our department whose interests are primarily computational often find it useful to take courses in the computer science department, where they can learn C and other computer science topics that we don't cover.

AI and Cognition

To the degree that there are models of cognitive processes founded in artificial intelligence that can claim both psychological and neurophysiological plausibility, then the answer to the title question of this essay is clear — cognitive science majors should know about them. And if the participants in this symposium know of such models, I am eager to incorporate them into our curriculum.

However I feel that traditional artificial intelligence has lost whatever claim it might have had to provide serious cognitive models. I understand that this claim is provocative, and I don't want the discussion to devolve into arguments for and against, but there are at least two reasons to be adduced in support of it, and both are relevant to the treatment of AI in the cognitive science curriculum.

The first reason to suspect that artificial intelligence can't provide the foundation for cognitive modeling is its deliberate ignoring of neurally plausible implementation issues. Graduate students in the 70s and early 80s were often taught that Turing universality made it possible to at least postpone such questions, that there was a relatively cleanly separable "knowledge level" where the important events of cognition occurred, and that attempting to understand the brain by looking at the activity of individual cells or electrical signals was like trying to understand a computer by measuring voltages on individual wires or by recording the electrical fields surrounding the CPU.

One result of this deliberate avoidance of such issues resulted in models which (however else they were successful or failures) could not be associated with the systems and mechanisms in the brain that were being worked out by neuroscientists using the very tools the artificial intelligence practitioners disdained. Another result was that some of the most successful artificial intelligence systems behaved in ways that were not even supposed to correspond to human mental operations.

The second reason to suspect that artificial intelligence can't provide the foundation for modeling cognition is that it is no longer the only game in town. Those who still do cognitive modeling in an artificial intelligence framework are called upon to compare their models with qualitatively different approaches, from neural networks, to other statistically-based approaches, to models based on real neural systems, or models inspired by dynamical systems theory. When traditional artificial intelligence was the dominant approach, any success at all in modeling a cognitive task was taken as presumptive support for the reality of the model. However the current plethora of approaches, many of which have successfully modeled cognitive phenomena as well as, or better than, artificial intelligence, makes clear the difficulty of arguing from the success of a model to a claim that minds really work that way.

The only known cognitive entities crucially involve brains and nervous systems. Traditional artificial intelligence was committed to the proposition that that fact could be ignored, at least for a while. It is possible that artificial intelligence is simply not old enough yet to really assess the validity of that commitment (or the field as a whole). However my own opinion (and I speak with varying degrees of support from the members of my department) is that an understanding of cognition can only be achieved by looking at the brain and by looking at behavior, and by building and studying computational models that respect the details of both.

Cognition and Computation

So where does that leave us? How much AI does a cognitive science major need to know?

One point to make immediately is that cognitive scientists need to have a good understanding of computation in general. If one of the fundamental claims of cognitive science is that the mind is somehow computational, then it follows that cognitive scientists need to know about computation. But "computation" isn't really a subject matter aside from the body of computational practice that has developed over the last half century. Theoretical approaches (such as Turing machines, formal semantics of programs, etc.) capture only part of the essence of computation. Most of the rest is embodied in a set of intuitions, standard examples, and experiences that the members of the programming community share. And it is from these intuitions — not from any of the specific formal models intended to capture them — that the fundamental grounding of cognitive science on computation rests.

To the degree that a specific historical artificial intelligence-based cognitive model is to be taught, it is important that the student understand the reality of the model, which is to say, the actual program, rather than what the experimenter claims the model shows, because what the model shows is something about the nature of computation itself, and is only about cognition via an interpretation step.

The only way to achieve that intuitive understanding of computation is to program. Students should be intimate with the details of what their computers can and cannot do, and the only way to achieve such intimacy is to spend a few nights together, puzzling through a program that seemed so clear when it was described on the assignment sheet.

If this is right, part of the training of a cognitive science must involve programming. (There are other,

practical, reasons for computer literacy also, as virtually all aspects of scientific life these days, from collecting data to discussing results to publishing papers, involves using computers.) Furthermore, since the idea is to instill intuitions, the cognitive science student should be exposed to a wide a range of programming paradigms as possible.

So my answer is that cognitive science students should see those aspects of AI that involve interesting and important computational paradigms, especially those that have been applied to cognitive modeling.

What are those? The following list is based on what I personally find the most valuable, but I hope the justifications make it clear what additional topics might have a place.

Symbolic Programming. Illustrated by programs that do calculus and algebra and natural language processing, among others. The key idea here is how the computer can be used to represent the symbolic structures in domains which are inherently symbolically structured. It is also important to point out that not all domains are like that.

Search. It is important for students to understand how general search techniques behave in various kind of search domains. The idea of heuristics can be introduced here, as can a discussion of complexity and the differences that changes in representations can make.

Structured Knowledge Representation. Object-oriented programming has made it into the computer science mainstream. It is thus important for students to understand object oriented programming for practical purposes. But the idea of organizing a program (or knowledge) in a set of objects, with their behavior organized as methods involving those objects, is straightforward and powerful and just might have some neurophysiological relevance.

Declarative Knowledge Representation. Logic is probably far from how people think, yet systems which use logical formalisms can be very expressively powerful. It is important for students to understand the expressivity/complexity/cognitive reality tradeoff.

Constraint Propagation. Again, constraint-based formalisms are finding their way into the mainstream. But the idea of a set of independent knowledge sources, each contributing a partial solution, combined via propagation to a global solution is very important. It is also important for students to understand the different views of this issue that are possible, everything from Waltz line-labeling to truth maintenance systems to Boltzman-machine connectionist networks.

In addition, I feel that the following "non-AI" topics are important:

Computer Architecture. Students should be able to understand the interface between computation and physics. They should understand how logic gates can be combined to make a computer. After all, in their neuroscience courses, they are expected to understand how the partial knowledge we have of neurons is related to the partial knowledge we have of brains. For computers, we have complete knowledge since we designed them, so they provide a good illustration of how at least one such story could go.

Parallel Programming. The brain is a parallel computer. Cognitive science students need to understand parallelism. Whether any of the current models of parallelism have anything to do with how the brain works remains to be seen, we need to train intuitions, and the existing models are all we have to work with.

Conclusion

My general argument has been: Cognitive science students need to be trained in programming so as to develop the intuitions about the nature of computation that they will need in order to understand how the brain implements cognitive processes. At this point in the development of the field we can't be so sure that any specific aspect of computation will turn out to be irrelevant. Students should thus be given a wide-range of programming experience. Artificial intelligence models, taught as programming exercises and situated within a large set of approaches to cognitive modeling, can provide part of this experience.