Teaching Introductory AI From First Principles

Susan L. Epstein and Virginia Teller
Department of Computer Science
Hunter College and The Graduate School of the City University of New York
695 Park Avenue
New York, NY 10021 USA
sehhc@cunyvm.cuny.edu

Abstract

Hunter College teaches introductory AI as a one semester, advanced undergraduate elective that focuses on issues in three areas: state-space search and control, knowledge representation, and Lisp. The search component defines AI methodology and conveys how the computing world views the field; knowledge representation illustrates how AI sees the real world; and Lisp shows how to make AI happen. The history and format of the course as it has evolved over a period of fifteen years is covered here, along with a description of the course content today and a rationale for teaching AI from this perspective.

1. Background

Hunter College is one of nine predominately undergraduate senior colleges of the 21-campus City University of New York. It is an urban, public, coeducational, liberal arts institution with an enrollment of more than 20,000 (about 15,000 FTE). Over 70% of the student body are women, and more than half are members of racial or ethnic minorities.

The Department of Computer Science was created in 1981 with five faculty members from the Department of Mathematics and an outside chair. Today the faculty of three women and seven men has research interests that include graph theory, combinatorics, software engineering, networks, and artificial intelligence. Each year the department enrolls about 1000 students in undergraduate courses and graduates roughly 45 of its 130 majors, 36% of whom are women and 42% minorities. Many computer science majors are older students returning to school for a first degree or seeking a second degree to switch careers. It is not uncommon for students with full-time jobs and family commitments to register for four courses a semester and attend classes four evenings a week for three hours.

2. Format

Since the early 1980's, AI has been taught at least once a year as a one-semester elective for advanced undergraduates. The texts used have included Patrick Winston's *Artificial Intelligence* (first and second editions) and Elaine Rich's *Artificial Intelligence* (first and second editions). Several options have been tried in developing the programming component of the course: no programming at all, the rudiments of both Lisp and Prolog, Prolog only, and Lisp only. The class has had access to a variety of IBM-compatible PC's and Macintosh computers. The versions of Lisp we have used have evolved along with the microcomputers: public domain Lisp for the Apple, IQ Lisp and GC Lisp for the PC's, and, most recently, Macintosh Common Lisp (MCL). When MCL arrived, Timothy Koschmann's *The COMMON LISP Companion* replaced Winston and Horn's LISP as the programming text.

In recent years AI has been offered yearly in the fall with Epstein and Teller teaching in alternate years. Rich and Knight's *Artificial Intelligence* (second edition) and Koschmann's book on Common Lisp are both required for the course. The class typically meets for two 75- minute sessions per week during a 14-week semester. After two or three introductory lectures, Lisp instruction begins in the Macintosh lab, either in a block of six or more consecutive classes or alternating with classroom lectures. As the instructor lectures and demonstrates, the screen on her computer is projected for the class to see, and the students can replicate the demonstration at their workstations. Unless the class is unusually large, this setup generally makes it possible for each student to gain hands-on experience during the lab; however students sometimes have to share computers. (Enrollment in the course is limited to 35, but the number is often smaller.)

3. Why Fundamentals?

There are a number of reasons for our decision to concentrate on first principles, rather than to offer up a smorgasbord of topics in introductory AI. Most important, we believe that without a firm grasp of the underpinnings, students cannot appreciate the accomplishments of the field. Viewed as a smorgasbord, AI may look like a conglomeration of artifacts, each one engineered to solve a different, specific problem. The commonalities among problem solving techniques are readily buried beneath the superficial dissimilarities.

Hunter students interested in pursuing AI after the introductory course can opt for undergraduate and graduate seminars on specific topics and application areas. We both teach advanced courses in our own areas of specialization -- Epstein in machine learning and knowledge representation, and Teller in natural language processing and computational linguistics. These topics cannot receive the in-depth treatment they deserve in one introductory semester. Moreover, an attempt to introduce AI with a limited overview of such topics risks leaving students with a peculiar perspective. Throughout the course, artifacts like AM and Dendral are used as hooks for topics and techniques in relevant areas.

4. What Is Fundamental?

Our approach to teaching introductory AI embodies the notion that the goal of the field is to build computer systems that can solve problems that are hard for computers but easy for people. Although cognitive problem solving techniques are stressed throughout the course, the emphasis is on simulating human intelligence rather than cognitive modeling or emulation. The textbook definition that most closely reflects our position on the importance of engineering workable solutions is found in Rich (1983, p. 37) and Rich and Knight (1991, p. 44), where AI is described as "the study of techniques for solving exponentially hard problems in polynomial time by exploiting knowledge about the problem domain." To convey this view of AI in an introductory course, the semester is organized into three nearly equal segments devoted to issues of search and control, knowledge representation, and Lisp, each of which is discussed further below.

4.1 Search and control

We use the state space paradigm as the framework for problem solving. When students see an interesting task as search in a very large space, they encounter a clear example of intractability. When they attempt to produce a clear definition of a goal state, they confront the subtleties of matching and of problem definition. When they attempt to maneuver through a large space, they intuit some of the standard control devices and learn about others. And, when these general methods fall short of any goal, students recognize the need for heuristics in a large search space.

We draw upon a wealth of examples for this material. In the first homework assignment, students record protocols of themselves solving several classic AI problems (e.g., the water jugs, Tower of Hanoi, cryptarithmetic), with an emphasis on the method rather than the answer. Throughout the semester their own experience offers clear demonstrations of many important points. Other favorite examples of ours are everyday behavior in very young children, playing board games, going home after class, and medical diagnosis. We also encourage students to assemble a vocabulary of classic AI artifacts from their reading assignments, and to cite them as they are relevant. In this manner, cognitive tasks are perceived as engineering feats where human problem-solving techniques can make significant contributions.

4.2 Knowledge representation

Examination of AI artifacts motivates the need for domain-specific knowledge and highlights the tradeoff in AI between generality and power. Through examples, we show the potential impact on search of knowledge and of a well-chosen representation for it. Although we devote some time to production rules, frames, and scripts, most of this section emphasizes predicate calculus. With this foundation we can go on to critique logic as a representation and, as time permits, to raise interesting issues for any form of knowledge representation: representational adequacy, inferential efficiency, transparency, granularity, and internal consistency.

4.3 LISP

Our work in LISP is motivated by a discussion about the nature of languages that best lend themselves to work in AI. We begin with objects as a way to represent state spaces, and methods as ways to move through them. As we introduce a basic vocabulary of LISP reserved words, we encourage our students to experiment through the interpreter and to write very small (20 to 30 lines) functions that call each other. (Some purchase Guy Steele's *Common Lisp - The Language* and explore it extensively.) In the last few years, we have designed the programming assignments as a series of increasingly sophisticated modules that eventually simulate some intelligent behavior, for example, play an arbitrary game or answer questions about a piece of text. Students demonstrate proficiency in coding and are expected to match and search in a well-understood domain. We can imagine no better way to confront the challenges and excitement of AI research than to build a program that really "looks smart. "

5. Syllabus.

Lectures cover most of the material in the first five chapters of Rich and Knight. These include all of Part I: Problems and Search and the first two chapters of Part II: Knowledge Representation as enumerated below:

I. Problems and Search
- 1 What is Artificial Intelligence
- 2 Problems, Problem Spaces, and Search
- 3 Heuristic Search Techniques

II. Knowledge Representation
- 4 Knowledge Representation Issues
- 5 Using Predicate Logic

As time permits, selected material from Chapters 9 through 12 may also be introduced. These topics include weak and strong slot-and- filler structures, a summary of knowledge representation, and game playing. A midterm test and a final exam on this material constitutes about 50% of the course grade.

The Lisp laboratory covers topics in the first six chapters of Koschmann including list primitives, logical operators, predicates, flow control function definitions, recursion, iteration, and CLOS. Some time is also spent on the MCL editor and debugger. The programs, plus additional homework (e.g., the protocols and logic exercises) constitute the remaining 50% of the course grade.

Additional books on artificial intelligence and Lisp, some current and others of primarily historical interest, are on reserve in the library. These include The Handbook of Artificial Intelligence and The Encyclopedia of Artificial Intelligence. A report on the robot competition at AAAI, complete with photographs, is a popular handout.

6. Issues

The strategy we have adopted for teaching introductory AI is not without its difficulties. Because we are unwilling to compromise on a single textbook that teaches AI without Lisp or Lisp without AI, and can find no satisfactory one that does both, students must buy two textbooks at a cost of nearly $100. This is undoubtedly a financial burden for some members of the class. The sheer amount of work, with deadlines every week, discourages other students, who perform poorly in part because they are unable to turn in assignments on time. As a result, one quarter to one third of the class is likely to drop during the course of any given semester. In addition, inadequate weekend hours and overcrowding in the Macintosh laboratory put evening students with full time jobs at a disadvantage. This is particularly serious because few students have access to a comparable version of Common Lisp elsewhere. We remain convinced, however, that ours is an undertaking well-worth doing. Every year we force students to think about the nature of human intelligence and challenge them to reproduce some fragment of it on a machine. And every year we addict at least a few to the challenges and excitement of AI, challenges they go on to pursue in their education and careers.

References

Koschmann, Timothy. 1990. The COMMON LISP Companion. New
York: Wiley.

Rich, Elaine. 1983. Artificial Intelligence. New York: McGraw-Hill.

Rich, Elaine and Kevin Knight. Artificial Intelligence, Second Edition. New York: McGraw-Hill.