

A Syllabus for Introductory AI

Matthew L. Ginsberg

CIRL

1269 University of Oregon

Eugene, Oregon 97403-1269

ginsberg@cs.uoregon.edu

1 Overview

Frankly, I don't see the problem. I have taught introductory AI at Stanford and although the course in general has a terrible reputation, it was a success both times I taught it. (One of the anonymous comments appearing in the campus paper was, "If you have to take this course, take it from Ginsberg.") This paper describes the organization of my course and also touches on a variety of other issues.

Let me say at the outset that Stanford in particular has a problem with intro AI classes; they have tended to be far too "Stanford-centric", focussing on first-order logic, occasionally to the exclusion of all else. When I taught the class, my focus remained on the formal side of AI (the neat view as opposed to the scruffy one), but I worked to extend the syllabus to include search, other methods of KR, and implemented AI systems.

There were three things that I believe made the course a success. The first was the fact that I could use "formal methods" as a backdrop against which all of the course could be set. The second was the way in which I integrated programming into the class. And the third was that I didn't use the course as a vehicle for promoting my own work.

With regard to a uniform focus on formal AI, my attitude was to make sure that I tied less formal work (e.g., NLP) into the formal stuff. I made clear the connection between search and parsing, and discussed semantic processing from the point of view of producing output that could be interpreted by a first-order theorem prover. I discussed ATMS's from a fairly formal point of view, and when it came time to discuss speedup learning, was careful to tie that back to the ATMS work. The discussion of frame systems included a section viewing them as resolution control strategies. And so on.

This technique allowed me to overcome the smorgasbord problem; the cost in terms of bias was only that I was committed to a neat view of AI. In my case, at least, this is a commitment that I was happy to make; it also seems appropriate for a computer science course (as opposed to a cognitive science approach).

As far as programming goes, I required that students know LISP when they began the class. (Or else be willing to learn it on their own.) There was no programming in the lectures, but the final exam involved writing a computer program to construct crossword puzzles. The students broke into teams of up to three members, all using LISP on identical Macintoshes. They were given 20 empty frames of various sizes and their programs were given three hours to fill in words (there were no clues, of course; it was purely a search problem). Each correctly completed frame was worth 5 points. I informed them at the beginning of the "exam" that my program, written in LISP and running on identical hardware, had completed the exam in three minutes.

There were a range of advantages of running the final this way:

1. The "crossword puzzle problem" provided yet another focus for the lectures. Any time I had a topic that I could tie back to it, I did. This happened frequently during the discussion of search, and also during the discussion of control of reasoning.
2. I didn't need to worry about programming either during the lectures or during the regular homework assignments. AI is still a science of *ideas* and it was a pleasure to be able to focus on that.
3. The grades on the final were available immediately. Within half an hour of the end of the final, every student knew what his grade in the class was. I could then file the grades with the registrar and go on vacation.
4. The atmosphere surrounding the final was cooperative. When I announced that, "In the spirit of AI, the final exam in this class will be taken by your computer programs," the reaction was completely positive.

The last broad point I would like to make is that it seems to me to be crucial to avoid using an intro AI class to promote one's own research, debug one's own software, and so on. This does absolutely no one a service, but seems all too common. There are far too

many examples of using textbooks to publish one's otherwise unpublishable research results or of using intro AI classes to promote a particular and specific research paradigm. For my part, I do not believe that I even *mentioned* my work on multivalued logics in the entire course.

2 Responses to specific questions

Core of a good intro course I view search and knowledge representation as the core of any good AI course. The other topics I chose to include can be found in the syllabus; topics I spent very little time on included robotics and connectionism. The basic reason was that I had plenty of other stuff to talk about and these topics fit very poorly into the overall framework of the class. Connectionism fit (sort of) into the discussion of learning.

Technique-based vs. problem-based The first part of the course was technique-based, as I discussed search methodologies, KR schemes and so on. The second part of the course was problem-based, focussing on planning, vision, NLP etc.

Curriculum details The course was part of a computer science curriculum, required for masters students and majors in intelligent systems. It was a one-quarter course taken by 3rd and 4th year undergraduates and 1st year graduate students. I required LISP and an understanding of formal logic as prerequisites.

With regard to a graduate/undergraduate distinction, I did not feel it necessary to make one. Most areas of AI (vision is one exception) still have the property that the problems and techniques are accessible to a broad audience.

3 Syllabus

The following syllabus is now the table of contents for my intro text.

I. INTRODUCTION AND OVERVIEW

1. Introduction: What is AI?
 - Defining artificial intelligence
 - What AI is about
 - What AI is like
2. Overview
 - Intelligent action
 - Search
 - Knowledge representation
 - Applications: examples

II. SEARCH

3. Blind search
 - Breadth-first search
 - Depth-first search
 - Iterative deepening
 - Iterative broadening
 - Searching graphs
4. Heuristic search
 - Search as function maximization
 - A*
 - Extensions and IDA*
5. Adversary search
 - Assumptions
 - Minimax
 - Alpha-beta search

III. KNOWLEDGE REP: LOGIC

6. Introduction to KR and logic
 - A programming analogy
 - Syntax
 - Semantics
 - Soundness and completeness
 - How hard is theorem proving?
7. Predicate logic
 - Inference using modus ponens
 - Horn databases
 - The resolution rule
 - Backward chaining using resolution
 - Normal form
8. First-order logic
 - Databases with quantifiers
 - Unification
 - Skolemizing queries
 - Finding the most general unifier
 - Modus ponens and Horn databases
 - Resolution and normal form

- 9. Control of reasoning
 - Resolution strategies
 - Compile-time and run-time control
 - The role of metalevel reasoning in AI
 - Runtime control of search
 - Declarative control of search

IV. KNOWLEDGE REP: OTHER TECHNIQUES

- 10. Assumption-based truth maintenance
 - Definition
 - Applications
 - Implementation
- 11. Nonmonotonic reasoning
 - Examples
 - Definition
 - Computational problems
 - Final remarks
- 12. Probability
 - MYCIN and certainty factors
 - Bayes' rule and the axioms of probability
 - Influence diagrams
 - Arguments for and against probability in AI
- 13. Frames and semantic nets
 - Introductory examples
 - Extensions
 - Inference in monotonic frame systems
 - Inference in nonmonotonic frame systems

V. AI SYSTEMS

- 14. Planning
 - General-purpose and special-purpose planners
 - Reasoning about action
 - Descriptions of action
 - Search in planning
 - Implementing a planner
- 15. Learning
 - Discovery learning
 - Inductive learning
 - Explanation-based learning
- 16. Vision
 - Digitization
 - Low-level processing
 - Segmentation and the Hough transform
 - Recovering 3-D information
 - Active vision
 - Object and scene recognition

- 17. Natural language
 - Signal processing
 - Syntax and parsing
 - Semantics and meaning
 - Pragmatics
 - Natural language generation
- 18. Expert systems
 - Examples and history
 - Advantages of expert systems
 - CYC and other VLKB projects
 - AI as an experimental discipline
- 19. Concluding remarks
 - Public perception of AI
 - Public understanding of AI
 - Applications of AI