The Introduction to Artificial Intelligence Show

Summer 1999

Professor Selmer Bringsjord
Dept. of Philosophy, Psychology & Cognitive Science
Department of Computer Science
Director, The Minds *&* Machines Laboratory
Rensselaer Polytechnic Institute
Troy NY 12180
(518) 276-8105 (P, Office) • (518) 276-4849 (P, Lab) • (518) 276-4871 (F)
Office Hours: Wed 10:30-12n
selmer@rpi.edu • http://www.rpi.edu/∼brings
Course Web site: http://www.rpi.edu/∼brings/intai.html

5/18/99

1 General Orientation

This course follows Norvig and Russell (the authors of our main textbook) in introducing artificial intelligence (AI) through the design and building of intelligent agents, which now routinely turn up in the marketplace. It includes introduction to, and significant development in, Common Lisp (some hands-on programming in theorem provers is provided as well), the programming language traditionally associated with AI in the United States. Please note that Lisp is a "real-world" language for "real world" problems and tasks. NASA's "Deep Space 1" probe, for example, is controlled real time by Lisp, with Earth-based engineers able to bring up down here a Lisp prompt allowing them to control DS1 in space.

This course seeks to explain AI as a set problems, techniques, formalisms, algorithms, and engineering objectives; AI is not the same as, nor is it to be inseparably joined with, any particular programming language. Though Common Lisp is the tradition, in theory any Turing-complete language could be used, and in fact Norvig and Russell do offer sample code, at the web site for their text, in other languages, including Java. In addition, any of the three main programming paradigms — functional, declarative, procedural (or, as Shapiro, the author of our Lisp textbook, says, "imperative") — could be studied to the exclusion of the other two in an introductory AI course. We will touch upon all three. Bringsjord does happen to prefer the declarative paradigm.

Some of the instructor's research in AI will be interjected into the course material.[1]

For on-site students, class discussion is encouraged; attendance is mandatory.

[1] E.g., We will go beyond R&N in the areas of expert systems; computational creativity (the agent BRUTUS$_1$, created by Dave Ferrucci (a scientist at IBM's T.J. Watson Research Center will be covered) and Bringsjord; deductive reasoning, theorem-proving, and diagrammatic reasoning; and the connection bewteen Gödel's first incompleteness theorem and "Strong" AI, which may involve a look at analog chaotic neural nets.

Bringsjord is the Director of the Minds & Machines Lab at RPI, and elements of this lab, which is devoted to AI and the closely related fields of applied cognitive science and cognitive systems engineering, will be introduced. For example, our coverage of robotics will probably be "jazzed up" with actual robots from the M&M Lab (and so we may do some light programming in Interactive C).

2 Teaching Assistant

Our TA is Selim Gurun (`guruns@cs.rpi.edu`). His office is Lally 316. Office Hours: Tuesday-Thursday 11:00-12:00am. Tel: (518) 276-6476.

3 *Intro to AI* and Fun

If any course can be interesting and fun, it has to be *Intro to AI*. One of the reason why AI attracts so many smart and motivated people is that it is just fundamentally cool; it just is. AI of today was sci fi yesterday; sci fi today is AI tomorrow. HAL started out as purely a creature of (cinematic) fiction; today NASA aims at building an "immobot" (stationary AI that controls a large structure) with many of the same powers and abilities (but hopefully without the same defects). Not long ago some of the smartest humans on the planet said that a computing machine could never play grandmaster chess; today, arguably the best chess-player on Earth is Deep Blue. What Deep Blue has done is something people have dreamed about for centuries, in some cases unsure about whether such an accomplishment was even possible.[2] Arguably the Information Age is at bottom simply the ascension of AI, and the systems it produces. AI researchers of today are working toward cars that drive themselves (we'll take some special looks at this goal), reliable household robots, real-time translation telephones. In fact, a recent spate of books (including, e.g., *Robot*, from Carnegie Mellon's Hans Moravec, a roboticist there) claims that AI isn't that far away from creating robots that are as smart as people. These books are written by mainstream AI engineers, not lunatics. All of this adds up, I think, to interesting stuff.

4 Texts

We have two required texts,

- *Artificial Intelligence: A Modern Approach*, by Russell & Norvig 1995 Edition (= R&N);
- *Common Lisp: An Interactive Approach*, by Stu Shapiro (S),

from which the readings in the schedule below come. R&N is a *big* book: coverage of the entire text would require (the equivalent of) a full academic year. Our trajectory through the book obviously doesn't include all chapters, but it is a certainly an ambitious one: strive to stay up-to-date in the reading.

If you're new to Lisp, start immediately to work with (S): this text is set up to allow students to become reasonable "Lispers" on their own, at their own pace, as long as the interactivity encouraged by the book is achieved by the student. Lisp is by its nature interactive: there is no substitute

[2] Baron Wolfgang von Kempelen, in the 1760's, toured Europe as the Maizal Chess Automaton. The system was a turbaned marionette in a cabinet, nicknamed "the Turk." Inside was a tiny chess master. A century later, Alan Turing, the grandfather of computer science and AI, inaugrated the path to Deep Blue: he developed a program that could generate simple moves and evaluate positions.

for sitting down with Lisp and experimenting. (Indeed, by Shapiro's lights, Lisp programs are written by sitting down and gradually evolving a program in interaction with Lisp.) Bringsjord's own coverage of Lisp will not stick closely to (S). He will presuppose most of the material in this text, rather than working from it (though my own code will refer to appropriate sections in (S).) Some parts of (S) are particularly relevant, others aren't; we'll discuss this. (E.g., packages will not be important for us, though they could be crucial in some contexts.)

If you become serious about Common Lisp and AI, two books should find their way into your library, viz.,

- *Common Lisp: The Language*, 2nd edition, by Guy Steele. (Actually, this text is recommended for the present course. It provides the complete and definitive syntax of CL.)
- *AI Programming: Case Studies in Common Lisp* by Peter Norvig.

As to texts on Prolog, which has been intimately associated with AI in Europe and Japan for many years, should you become interested in this powerful language, the following texts are recommended.

- *Programming in Prolog* by Clocksin & Mellish (4th edition, can be ordered off the web);
- *The Art of Prolog* by Sterling & Shapiro (a classic text);
- *Foundations of Logic Programming* by Lloyd (a must for truly understanding Prolog).

5 Computing

A mere networked personal computer is sufficient for this course. All students will need access to W^3, as the course web site is at the very heart of the course. (E.g., all class slides, all handouts, all practice exams, all demos, etc. will be distributed via the web site.) Demos and code will be available in simple ASCII; handouts and slides will generally be available in dvi, postscript, html, and pdf. Bringsjord will of course explain all of this.

All students will obviously need to have a Common Lisp package available to them. It must be *Common* Lisp. You can acquire a free version of Allegro CL (the "Lite") version from Franz Inc. — a link to this company can be found on the web site for this course. This package is for Win 95/NT. (A Linux version of ACL is also available for no charge.) Digitool Inc. offers an excellent Mac version of CL, but it's expensive (the link is on our site). There are a number of public domain CL packages floating around the Net; some are pretty darn stable. As a fallback, all students have access to Kyoto Common Lisp by telnet into RCS.

6 Schedule

Note:

- It seems all great shows have at least one lost episode. The Intro to AI Show is no exception: we do indeed have ... "The Lost Episode" — a supplementary lecture on the propositional calculus (Ch. 6 in R&N). Fortunately, this episode has been found. It will be made available in the library for the studio audience, and will be mailed to home viewers.
- Each class meeting results in two lectures, and hence two tapes.
- Our final exam comes during the two-day final exam period after normal course meeting times are over. The exact time of the exam for the studio audience will be set by the Registrar.

All readings in the following table refer to (R&N):

Date	Class#	Lec#s	Topic	Reading
5/18	1	1/2	Nature & History of AI	Ch. 1
5/20	2	3/4	The Agent Approach; Search; Games Project 1 out	Chs. 2, 3, 5
5/25	3	5/6	Logical Agents; FOL; KBs	Chs. 6, 7
5/27	4	7/8	Inference; Reasoning Systems Project 1 due; Project 2 out	Chs. 8, 9
6/1	5	9/10	Diagrammatic Reasoning; Review **Midterm**	Ch. 10.1-10.3, 10.4-10.9
6/3	6	11/12	Planning	Ch. 11
6/8	7	13/14	Uncertainty: OSCAR; creativity: BRUTUS$_1$	TBA
6/10	8	15/16	Learning I Project 2 due; Project 3 out	Chs. 18, 19.1-19.4
6/15	9	17/18	Learning *via* ILP; Zoombinis	Ch. 21
6/17	10	19/20	Communicating Agents	Ch. 22
6/22	11	21/22	Robotics; Math Objection; Future	Chs. 25.1-25.4,26, 27
			Final Exam (Project 3 due)	

7 Grading

There will be two exams, a midterm (25%) and a final (35%), as well as three programming projects (counting, from first to third, for 10%, 15%, and 15%). As an aid to studying, both a practice midterm and a practice final will be provided (on the web site). The grading of projects will be eminently fair: I don't expect students to become Lisp experts overnight. I'm especially sensitive to the challenge of shifting from a procedural (= imperative) mindset in programming, to a functional or declarative mindset. There will occasional "pop problems" presented during lecture. Though these problems will not figure in grade calculation, students are required to try to solve them on the honor system. These problems are designed so that everyone can solve them given sufficient effort, and sufficient command over (R&N). Also, as part of the author's experimental research, students will be confronted with a number of puzzles (all of which make for interesting test cases for clever artificial agents). Please appeal grades by contacting me in person.

8 Academic Honesty

Student-teacher relationships are built on mutual respect and trust. Students must be able to trust that their teachers have made responsible decisions about the structure and content of the course, and that they're conscientiously making their best effort to help students learn. Teachers must be able to trust that students do their work conscientiously and honestly, making their best effort to learn. Acts that violate this mutual respect and trust undermine the educational process. They counteract and contradict our very reason for being at Rensselaer and will not be tolerated. Any student who engages in any form of academic dishonesty will receive an F in this course and will be reported to the Dean of Students for further disciplinary action. (The *Rensselaer Handbook* defines various forms of Academic Dishonesty and procedures for responding to them. All of these forms are violations of trust between students and teachers. Please familiarize yourself with this portion of the handbook.)