6.871 SPRING 2006 READING LIST: Installment #1

Lecture 1: *Course Organization; Spirit of Undertaking*

1. Feigenbaum, E.A., Knowledge Engineering. The Applied Side of Artificial Intelligence, Annals New York Academy of Sciences, 91-107.
   A call to arms that lays out the spirit of knowledge based systems, some of its promise, and goals.

2. Newell, A., The Knowledge Level, *Artificial Intelligence Magazine*, Summer 1981, pp. 1-20+.
   A somewhat theoretical statement whose intention is to demonstrate that it makes sense to talk about the performance of a program at the "knowledge level," i.e., in terms of what it knows, rather than in terms of what computations it carries out. This becomes particularly clear when Newell explains that we can talk about *what* an agent knows (i.e., its knowledge level description) without any idea about how that knowledge is *represented* inside the agent (i.e., its symbol-level description).

   This is a useful point to keep in mind as we go through the course: we want you to understand both what knowledge is needed and how to represent it, but we also want you to be able to distinguish carefully and know which one matters when.

3. Topolski A S, Reece D K, Packaging Advisor^{tm}: An expert system for rigid plastic food package design, in *Proc Innovative Applications of AI*, 1989, Schoor and Rappaport (eds.), pp. 348–357.
   This is one small, well-described application, to give you a feeling for what expert systems can be, and why they can matter. Note in particular what the company got from building the system; pay attention to the wide variety of benefits that resulted.

Lecture 2: *Tell it what to know; Search*

1. Korf, R. E., Search: A survey of recent results. *Exploring Artificial Intelligence*, Shrobe, H. E. (ed), Morgan Kaufmann, 1988, pp. 197-237, and *ACM Computing Surveys*, vol. 27, No. 3, Sept 1995, pp. 337-339 [an update to the previous paper].
   If you are somewhat lacking in AI background, you would do very well to read these two papers. We will often rely on concepts of search in the rest of the course, so you ought to know the basics, at least. You should read these closely enough to have a feel for what's hard and what's not. Instant recollection of particular techniques is not the goal.

2. Rowley, S., Shrobe, H., Cassels, R. and Hamscher, W., Uniform Access to hetrogenous knowledge structures, or Why Joshing is better than Conniving or Planning. In *AAAI-87*, pages 48-52.
   This is about a language which tries to support hybrid reasoning, i.e., an approach in which you can mix together many components supporting different paradigms. We use it as one example of what it means for a language to provide *direct support* for a representation and/or reasoning method.

Lecture 3: *Origins of KBS:* MACSYMA and DENDRAL

1. Slagel J R, A heuristic program that solves symbolic integration problems in freshman calculus, in *Computers and Thought*, pp. 191–203.
   Pay particular attention to the table on page 201. What does it tell you about the effect of the heuristics on solving the problem? In particular, how useful were they?

2. Moses J, PhD thesis, Chapter 2: How SIN differs from SAINT.
   In this very short piece of text from his PhD thesis Prof. Joel Moses (former MIT Provost) explains how his approach is substantially different in method and mindset from SAINT.

3. Feigenbaum, E.A., Buchanan, B.G., Lederberg, J., On Generality and Problem Solving: A Case Study Using the DENDRAL Program. in Meltzer and Michie (eds) *Machine Intelligence 6*, pp. 165-189.
   This article is in part about the balance of power in a program using generate and test. Consider what makes the generator gets smarter and what consequences this has for the tester.

Lecture 4: *Application Analysis Case Study: Case Introduction*

1. Sviokla J, Smartwave: The Wave-Soldering Expert System, Harvard Business School Cases #0–187–062 and #0–187–063
   This lecture will set the background for your analysis of the wave soldering system as a knowledge-based system. Use the notes from lecture as a guide to the kinds of things to look for in the problem description.

   This is a real problem, hence there is a lot of information in these descriptions; some of it is irrelevant to your needs; some of what you need may be missing. Your job is to *find what you need*, to *ask about what you need but don't have*, and to *do the best analysis you can*.

   You should concentrate on the first case, Smartwave (A), trying to determine to what extent it meets the criteria outlined in the lecture. Then look briefly at Smartwave (B), to see what sorts of problems they faced in the next step of development.

Lecture 5: *Application Analysis Case Study: Class Discussion*

1. *Written Assignment:* Prepare a brief written report on the case analysis to use in the class discussion. Your write-up should concentrate on Smartwave A and explain what aspects of the task made it a good candidate and what aspects made it a poor candidate.

   Your write-up should also provide a list and description of the problems Smartwave encountered at the next stage (case B). What problems did they face and why?

Lecture 6: *Rule Based Systems*

1. Davis, R., Production Rules as a representation for a knowledge-based consultation Program, *Artificial Intelligence Journal*, 8:15-45, 1977.
   What are the basic strengths and weaknesses of rule-based systems, as reported here?

2. Duda, R., Gaschnig, J., and Hart, P., Model design in the prospector consultant system for mineral exploration, *Expert Systems in the Microelectronic Age* Michie (ed.), Edinburgh University Press, 1979, pp. 154–167.
   Read this to see another way to implement a rule-based system, with a different approach to handling uncertainty.

3. Campbell A. N., Hollister, V. F., Duda, R. O., and Hart, P. E., Recognition of a hidden mineral deposit by an artificial Intelligence program, *Science*, 217:927-929, Sept. 3, 1982.
   This short piece is interesting documentation of the significant real-world impact knowledge based systems can have.

Lecture 7: *Semantic Nets*

1. Quillian, M. Ross, Semantic Memory. In M. Minsky (ed.) *Semantic Information Processing*, The MIT Press, 1968, pp. 227–270.
   This is the grand-daddy of semantic nets. Almost all the details are wrong if not outright silly. Don't worry about trying to commit this stuff to memory. Instead concentrate on spirit of this idea and the insights that lead in this direction.

2. MacGregor R, The evolving technology of classification-based knowledge representation systems, in *Principles of Semantic Networks*, Sowa J. (ed), Morgan Kaufmann, 1991, pp. 385–400.
   Read this to get the basics of the more recent uses of semantic nets, to understand the more well-defined role they have come to play in knowledge representation and reasoning.

3. Brachman R, et al., Living with Classic: When and how to use a KL-ONE like language, in *Principles of Semantic Networks*, Sowa J. (ed), Morgan Kaufmann, 1991, pp. 401-456.
   There is some overlap here in describing the machinery, but this paper is one of the few to attempt to make explicit the *when and how* part of knowledge representation. Pay particular attention to the nice analysis of when it makes sense to use this particular representational tool.

Lecture 8: *Logic: when the paradigm is deduction*

1. Sergot et. al, The British Nationality Act As A Logic Program, *Communications of the ACM*, May 1986, 370–386.
   This is one of the few well documented applications written in Prolog that use Prolog almost totally as a logic engine. Pay particular attention to why they thought Prolog was the right tool and what troubles they had with it.

2. Smith D, Parra E, Transformational approach to transportation scheduling, *Proceedings of the 8th Knowledge-Based Software Engineering Conference*, Chicago, IL., Sept. 14–17, 1993, IEEE Computer Society Press, p60-68.
   This program provides a truly striking success on the important real-world task of scheduling. It does not generate *schedules*, its job is to generate *scheduling algorithms*, and as such it is a knowledge-based program whose expertise lies in knowing how to match a search strategy to the characteristics of a search problem.

   As one example of the power of the system, they generated a scheduler for a realistic military transportation problem involving 10,000 movements. The resulting code took about 2 minutes to generate a schedule, where existing programs require roughly 36 *hours* to do the same thing.

   There is a lot of technical detail here but not all of it is crucial. You should pay particular to section 4.2, understanding the role of automated deduction in pruning the search space.

3. Brachman R, Levesque H, Undirected behavior without unbounded search, *ACM Surveys*, Sept 1995, pp. 314–316.

4. Ginsberg, M L, Epistemological and heuristic adequacy revisited, *ACM Computing Surveys*, Sept 1995, pp. 331-333.
   These two short papers challenge an important assumption underlying the logicist approach and (to some degree) this course, namely that we can in fact tell a program what to know, to some degree distinct from what it should do. The question is, how tightly connected are knowing *what* and knowing *how*?

Lecture 9: *Frame based programs*

1. Minsky M, A framework for representing knowledge, in *Readings in Knowledge Representation* Brachman, R., Levesque, H. (ed), Morgan Kaufmann, pp. 246-262.
   This is the original paper on the notion of frames and as such has large a number of holes in it, places where speculations are made but little is made specific. Get the overall view here, by focusing on two things. First try to understand the spirit of the message–about a certain style of reasoning and knowledge representation–without worrying about the details. Second, notice the sharp contrast Minsky sets up with the logicist approach; see the Appendix to the paper for a particularly sharp critique.

2. Pauker, S.G., et al Towards the Simulation of Clinical Cognition. Taking a Present Illness by Computer. In *Readings in Medical Artificial Intelligence*, Clancey, W., Shortliffe, E., Addison-Wesley, 1984, chap 6.
   The relevant point here is how the frame idea has been used in keeping with the original intention of the notion.

3. Miller, et al Internist-1, An Experimental Computer-Based Diagnostic Consultant for General Internal Medicine. In *Readings in Medical Artificial Intelligence*, Clancey, W., Shortliffe, E., Addison-Wesley, 1984, pp. 190-209.

Read this to see an example of how frames can be used at the knowledge level: Think about how using frames provides a view of the diagnosis problem that is very different from the one produced when using rules (e.g., MYCIN).