6.871: KNOWLEDGE BASED APPLICATIONS SYSTEMS
SPRING 1995

Instructor :	Randall Davis	NE43-801A	3-5879	davis@ai.mit.edu
Secretary:	Kathy LaFrance	NE43-801	3-8919	kathy@ai.mit.edu
TA:	Michael Bolotski	NE43-703	3-8170	misha@mit.edu

Prerequisites: 6.034, 6.824, equivalent course, or equivalent experience.

Work in the course has three main components:

- Readings: there are substantial readings from original literature that go with each lecture.
- Lectures: students are expected to have done the reading and to participate in class discussion.
- Term project: each student is required to build an expert system in a domain of your own choice. To ensure that the project is done on a timely basis, we require an initial proposal for the project and a progress note (see syllabus). We will comment on the project proposals to help ensure your choice is appropriate in character and scope.

  We encourage collaboration on the term project, in groups of size two.

There will also be some brief laboratory assignments and homework.

Grades in the course will be determined primarily by the term project (including your paper describing the project and your presentation of it), by your participation in class, and by performance on homework assignments.

6.871: COURSE OUTLINE — Spring 1995

Knowledge Based Application Systems
Randall Davis and Howard Shrobe

Lecture	Lecturer	Date	Topic
INTRO & BACKGROUND			
1	D	Tu Feb 7	Course Organization; Spirit of the Undertaking
2	D	Th Feb 9	Technical Background and Perspective
3	D	Tu Feb 14	Decision Support: Other approaches to assistance
THE SPIRIT OF KBS			
4	D	Th Feb 16	Origins of KBS: MACSYMA and DENDRAL
		Tu Feb 21	NO CLASS (MIT schedule weirdness)
5	D	Th Feb 23	Application Analysis Case Study: Case Introduction
6	D	Tu Feb 28	Application Analysis Case Study: Class Discussion
KNOWLEDGE REPRESENTATIONS			
7	D	Th Mar 2	Logic: When the paradigm is deduction
8	D	Tu Mar 7	Rule Based Systems: MYCIN, PROSPECTOR ***Class Project Proposal DUE***
9	D	Th Mar 9	Semantic Nets: Quillian, KLONE, NIKL, etc.
10	D	Tu Mar 14	Frame based programs: INTERNIST-I, PIP
11	D	Th Mar 16	Pragmatic Issues in Knowledge Acquisition
12	D	Tu Mar 21	Overview of Knowledge Representation
13	D	Th Mar 23	Challenging Issues in Representation and Reasoning
		Mar 28, 30	Spring Vacation
PROBLEM SOLVING PARADIGMS			
14	D	Tu Apr 4	Problem Solving Paradigms: Intro and Example
15	D	Th Apr 6	Diagnosis and Assembly: MYCIN, INTERNIST, ABEL ***Progress Report DUE***
16	D	Tu Apr 11	The importance of the right abstraction
17	D	Th Apr 13	Model-Based Reasoning: HT, GDE, SHERLOCK
		Tu Apr 18	Patriots' Day Vacation
18	D	Th Apr 20	Case-Based Reasoning
19	D	Tu Apr 25	Evolution of Programming Languages: direct support
RESEARCH ISSUES			
20	Guest	Th Apr 27	Learning
21	D	Tu May 2	Reasoning about physical systems
22	Guest	Th May 4	Uncertain Reasoning
PROJECT PRESENTATIONS			
23	you	Tu May 9	Project Presentation & Discussion
24	you	Th May 11	Project Presentation & Discussion
25	you	Tu May 16	Project Presentation & Discussion
26	you	Th May 18	Feedback: Last Day of Classes *** Term Project Report DUE***

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: *Technical Background and Perspective*

1. Korf, R. E., Search: A survey of recent results. *Exploring Artificial Intelligence*, Shrobe, H. E. (ed), Morgan Kaufmann, 1988, pp. 197-237.
   A good survey of the state of the art in search. You should read this closely enough to have a feel for what's hard and what's not. Instant recollection of particular techniques is not the goal.

Lecture 3: *Decision Support Technology: Other approaches to assistance*

1. Chapter 1 & 2, *DECISION ANALYSIS: Introductory Lectures on Choices under uncertainty* by Howard Raiffa, Addison-Wesley Press, 1970. pp. 1-38.
   An introduction to the basics of making decisions under uncertainty when you have some knowledge of what the payoffs will be. We don't expect you to memorize the technical details; you can always refer back to this article if you ever need them; read this deeply enough to get a feel for the approach. When you read this think about what the pragmatic difficulties are in getting the data required to apply the techniques.

Lecture 4: *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? 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 (now MIT Dean of Engineering) explains how his approach is substantially different in method and mindset from SAINT.

3. Moses, J., Algebraic Simplification: A Guide for the Perplexed, *Communications of the ACM*, 14:8, August 1971, pp. 527-537.
   Don't get perplexed trying to learn the algebraic techniques. The big lesson in this paper is about matching representations to algorithms. It's presented with somewhat whimsical political categories.

4. 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.
   Ths article is in part about the balance of power in a program using generate and test. Consider how the generator gets smarter.

Lecture 5: *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 6: *Application Analysis Case Study: Class Discussion*

1. *Homework:* Prepare a brief written report on the case analysis to use in the class discussion. Your writeup 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 writeup 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 7: *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. AMEX, *Proceedings of the first conference on Innovative Applications of Artiticial Intelligence*, Stanford, Palo Alto, March 1989
   This is one of the premier success stories for Expert Systems. It's still running today. Think about how this application is similar to and different from the Nationality act.

3. Artificial Intelligence Goes to Work, *High Technology*, April 1987, pp. 16–27.
   Primarily a marketing document directed toward industry, but it gives another slant on the Amex application.

Lecture 8: *Rule Based Systems:* MYCIN, PROSPECTOR

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 sytem for mineral exploration, *Expert Systems in the Microelectronic Age* Michie (ed.), Ediburgh 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 9: *Semantic Nets: Quillian,* KLONE, NIKL*, etc.*

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 sematic 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. Brachman, R. J., On the epistemological Status of Semantic Networks, In N. V. Findler (Ed.), *Associative Networks: Representation and Use of Knowledge by Computers.* New York: Academic Press.
   This paper attempted to make sense of many dozens of very different systems all of which call themselves "Semantic Networks". Try to figure out what the spirit of each type is.

3. To be added.

Lecture 10: *Frame based programs:* INTERNIST-I, PIP

1. 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.

2. Goldstein, I. P., and Roberts, R. Bruce, NUDGE, A knowledge-based scheduling program, *5th International Joint Conference on Artificial Intelligence*, 1977 pp. 257-263.
   Here's the beginning of the degeneration of frames into a notion that begins to look an great deal like object-oriented programming. What piece of machinery described here starts leading in that direction?

3. Fikes and Kehler, The role of frame-based representation in reasoning, *Communications of the ACM*, Sept 1985, pp. 904–920.
   Here the degeneration has proceeded even further. Can you spot the places where the spirit of the representation has been lost under the machinery?

4. 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 (ie, MYCIN).

Lecture 11: *Pragmatic Issues in Knowledge Acquisition*

1. McDermott, J., R1: The Formative Years., *Artificial Intelligence Magazine*, Summer 1981, pp. 21-9.
2. Bachant, J., McDermott, J., R1 Revisited: Four Years in the Trenches. *Artificial Intelligence Magazine*, Fall 1984, pp. 21-32.
3. van de Burg, A., Bachant, J. and McDermott, J., The Taming of R1, *IEEE Expert*, **1**:33-39 fall 1986.
4. Barker V, O'Connor D, Expert systems for configuration at Digital: XCON and beyond, *Communications of the ACM*, March 1989, pp. 298–318.
   All four of these show pragmatic techniques used in R1. Consider in particular what changed in the machinery between the first two papers here and the last two.

Lecture 12: *Knowledge Representation Overview*

1. Davis R, Shrobe H, Szolovits P, What is a Knowledge Representation? *AI Magazine*, **14**, #1, Spring 1993, pp. 17–33.
   The basic argument here is that a knowledge representation performs five important roles and that ignoring any one of them is a bad idea. What are the five roles and how well do they fit with any representation you have in mind?

Lecture 13: *Challenging Issues In Representation and Reasoning*

1. Davis, R., Expert Systems: Where are we? And where do we go from here? *Artificial Intelligence Magazine*, 3:2, 1982, pp. 3-22.
2. Davis, R., Expert Systems: How far can they go? *Artificial Intelligence Magazine*, **10**:1 pp. 61–67 (part one), and **10**:2 pp. 65-77 (part two).
3. Davis, R., Mycin: A Retrospective View, unpublished memo.
   These three explore different views of the limitations in the reasoning and representation technology seen in the course thus far.

Lecture 14: *Problem Solving Paradigms: Intro and Example*

1. Chandrasekaran, B., Towards a functional architecture for intelligence based on generic information processing tasks. *International Joint Conference on Artificial Intelligence*, 1987, pp. 1183–1192.
   This is one view of what we have been calling problem solving paradigms.
2. Clancey, W. J., Heuristic Classification, *Artificial Intelligence Journal*, **27**:289–350, 1985
   This paper is notoriously hard to read, but buried in there are some good observations about what MYCIN was actually doing with all those rules. Consider how different this description is, as compared with viewing the system as a collection of rules. Do your best with the paper; we'll also try to make sense of it in class.

Lecture 15: *PSP: Diagnosis and Assembly:* MYCIN, INTERNIST, ABEL

1. Patil, R. S., Szolovits, P., and Schwartz, W. B., Causal understanding of patient illness in medical diagnosis. *Readings in Medical Artificial Intelligence*, Clancey, W., Shortliffe, E., Addison-Wesley, 1984, pp 229–360.
   This is another approach to the diagnosis problem, which differs in substantial ways from the view supplied by Mycin's rule-based approach. Think about issues such as, What does this system know? How does it represent that knowledge? What is its approach to diagnosis?

2. Patil, R. S., Artificial Intelligence Techniques for Diagnostic Reasoning, *Exploring Artificial Intelligence*, Shrobe, H. E. (ed), Morgan Kaufmann, 1988, Chapter 9.
   This is a good overview of the diagnostic problem in all of its complexities, demonstrating that AI programs still have some distance to go in covering the problem.

3. Simmons, R, Davis R, Generate Test Debug, *International Joint Conference on Artificial Intelligence*, 1987, pp 1071–1078.
   An interesting variation on the traditional generate and test paradigm.

Lecture 16: *PSP: The importance of the right abstraction*

1. Hamscher, W., Temporally coarse representation of behavior for model-based troublshooting of digital circuits, *International Joint Conference on Artificial Intelligence*, 1989, pp. 887–893
   A case study in how the right level of abstraction in the representation can make all the difference.

2. Rich, C., Shrobe, H. E. and Waters, R. C., Computer aided evolutionary design for software engineering. MIT/AI Memo 506, Jan 1979.
   Amongst other things, this paper contains an argument for why engineers (and other problem solvers) must use abstractions and approximations and therefore why debugging is an inherent part of problem solving.

Lecture 17: *PSP: Model-Based Reasoning*

1. Davis, R., Diagnostic reasoning based on structure and behavior, *Artificial Intelligence Journal*, **24**: 347–410, Dec.84
   A detailed example of model-based reasoning at work, using representations of the sort described in the previous paper. It also shows how to deal with the difficult problems that arise when the structure description itself differs from the object in unknown ways.

2. Davis, R., and Hamscher, W., Model-Based Reasoning: Troubleshooting, *Exploring Artificial Intelligence*, Shrobe, H. E. (ed), Morgan Kaufmann, 1988, Chapter 8.
   A survey of the fundamental issues in model-based reasoning and suggestions about the research problems that are still open.

Lecture 18: *Case-Based Reasoning*

1. To be specified later.

Lecture 19: *Evolution of Programming Languages: direct support*

1. Sheil, B., Power Tools for Programmers, *Datamation*, Feb 1983, pp. 131-144.
   This paper is actually a bit tangential, but it does contain a nice overview of the kind of programming environments that AI people build to help themselves solve hard problems. Many of the "high end" tools provide similar facilities; many of the "low end" tools don't.

2. Winograd, T., Chapter 6&7, *Understanding Natural Language*. pp. 108–126.
   This is the part of Winograd's thesis on the SHRDLU natural language understanding system which describes the blocks world problem solver. This was the first application built in Planner, and one of the last. We want you to think about two things while reading this lightly: 1) What is problem solving paradigm 2) How well does the Planner language support it?

3. Stallman, R. M., and Sussman G. J., Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis. *Artificial Intelligence Journal*, **9**:135–196, 1977.
   The back of this paper contains a detailed trace of the program which runs for about a dozen pages. Don't waste time on that. The first part describes a problem solving paradigm and the language mechanisms (dependency networks) that support it. Don't worry too much about understanding the electronics; focus instead on the problem solving paradigm.

4. 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.

Lecture 20: *Learning: An Update*

Lecture 21: *Reasoning about physical systems*

1. Forbus, K. D., Qualitative Physics: Past, Present and Future. Chapter 7, *Exploring Artificial Intelligence*, Shrobe, H. E. (ed), Morgan Kaufmann, 1988.
   This is an overview of work on the area called Qualitative Physics. The earlier part of the paper is good, but there are points later on where it gets obscure. The discussion about landmark values is obscure and the section on qualitative kinematics is also hard to follow.

2. To be augmented.

Lecture 22: *Uncertain reasoning*

1. Pearl, J., Evidential Reasoning Under Uncertainty, Chapter 10, *Exploring Artificial Intelligence*, Shrobe, H. E. (ed), Morgan Kaufmann, 1988.

Lecture 23: *Project Presentation & Discussion*

Lecture 24: *Project Presentation & Discussion*

Lecture 25: *Project Presentation & Discussion*

Lecture 26: *Feedback: Last Day of Class*