

Computational Logic: CS 6374

# Suggestions for Projects

Instructor: Gopal Gupta

Some suggestions for projects are listed below (see also part IV of the book). In lieu of the project, you can pick up any year's Prolog programming competition and program all but one problem (except 2003 competition). Prolog programming competition problems can be found at:

<http://www.cs.kuleuven.ac.be/~bmd/PrologProgrammingContests/>

## Suggestions:

1. A small rule-based expert system (such as trouble-shooting in cars).
2. A itinerary advisor, that will draw up an itinerary for a tourist given some budget constraints, and interests of the tourist, plus other information.
3. A compiler for a subset of a language (say Pascal); see Ch 24 of the text book).
4. Some scheduling problem (for example, keeping track of all hardware/manpower that a military has along with its location etc. and then given a request (during wartime, say) finding an optimal way to satisfy that need.
5. A program for playing a game (e.g. Othello, finishing positions in Chess, Kalah).
6. Figuring out when an undergraduate or graduate student in Computer Science will be able to graduate. This program could work in 3 ways: (i) Given a schedule it should be able to tell whether the student is eligible to graduate; (ii) Given a partial schedule it should be able to tell all possible ways in which a student can graduate (perhaps taking student preference into account); (iii) Tell a new student (who has decided his/her major) all possible schedules that will lead to graduation. This might be a real useful tool, if implemented.
7. Developing an intelligent TA to course matching system for CS at UT Dallas. Given a list of students and courses they have taken, find a TA assignment for each course

(the student must have taken the course they are assign to TA). Student preferences should be taken into account (students will give a preference order for the courses they wish to TA). Likewise, faculty preferences should be taken into account (they might prefer one student over another).

8. A formal or natural language translation system. For example, translating a small subset of English to Spanish (or any other language). Translating one database language to another. Writing a compiler for a small language. Definite Clause Grammars will have to be used in the parsing phase. (You should know definite clause grammars for this).
9. A system for aircraft collision avoidance.
10. Any intelligent WEB application (e.g., a comparison shopping tool). A Prolog Library for manipulating HTML pages is available and will have to be used in this project.
11. Just about anything else that you can think of. Use your imagination! See also chapters in Part III of the text book.