EECS3401 Introduction to AI and Logic Programming
Syllabus (tentative)

Course Objectives & Content

This is from the description from the academic supplemental calendar.

Artificial Intelligence (AI) deals with how to build systems that can operate in an intelligent fashion. In this course, we examine fundamental concepts in AI:

- knowledge representation and reasoning,
- search,
- constraint satisfaction,
- reasoning under uncertainty, etc.

The course also introduces *logic programming*, a programming paradigm based on predicate logic, where one specifies problems in a declarative way and one can use the language to search for a solution. Students will learn how to develop programs in **Prolog** to solve AI problems.

The course covers the following topics:

- Introduction to Artificial Intelligence, intelligent agents.
- Logical representations, first-order logic syntax and semantics, use in knowledge representation.
- Basics of logic programming and Prolog, syntax, backward chaining.
- Inference in first order logic, unification, resolution.
- Prolog control flow, backtracking, closed world assumption, negation as failure.
- Prolog lists, arithmetic.
- Uninformed search.
- Informed search.
- Constraint satisfaction and backtracking search.
- Planning and heuristic search.
- Uncertain reasoning, Bayesian Networks.

Prerequisites (may not be up-to-date, please confirm with EECS as needed)

P1 *general prerequisite* AND
P2 **(EECS-2011** OR **EECS-2101** OR **EECS-2030**) AND
P3 **MATH-1090**.

Note: The *general prerequisite* is a cumulative GPA of 4.50 or better over all major EECS courses. (EECS courses with the second digit "5" are not major courses.)

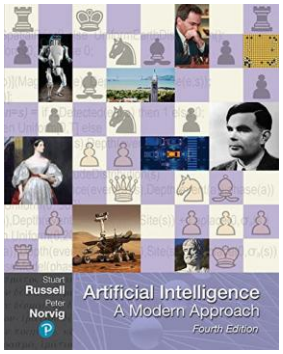
Learning Outcomes (*from the academic supplemental calendar*)

By the end of the course, the students are expected to be able to:

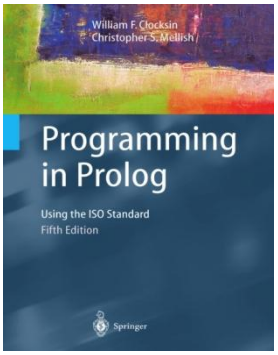
- Define the main objectives of artificial intelligence.
- Describe how first-order predicate logic forms the basis of logic programming.
- Write logic programs in Prolog.
- Use and modify heuristic state-space search algorithms such as A*, RTA* and IDA*.
- Represent knowledge in a small domain using predicate logic and use the representation to build a logical database for a knowledge-based expert system.
- Describe and use some other AI techniques including backward and forward chaining, Bayesian networks, game search and constraint satisfaction, grammars for natural language processing.

Textbooks for the Course

There are two required textbooks for the class: the first is a comprehensive textbook on AI; the second on the programming language Prolog.



Book title : Artificial Intelligence: A Modern Approach
Authors : Stuart Russell & Peter Norvig
Date : 2020 (4th edition)
Publisher : Pearson
ISBN-10/-13 : 0134610997 / 978-0134610993



Book title : Programming in Prolog: Using The ISO Standard
Authors : William F. Clocksin & Christopher S. Mellish
Date : 2013 (5th edition)
Publisher : Springer
ISBN-10/-13 : 3540006788 / 978-3540006787

This Prolog book is available electronically via the York University Libraries to students, staff, and faculty: *Programming in Prolog* (via EZProxy through York University Libraries). To access this, you need to be coming from a computer under the York University domain. When you are off-campus, consider using EECS's Remote Lab. See Login and Remote Access [EECS Technical Database] for information about logging into EECS computer services from off-campus. (The Library has licences for around 250 concurrent accesses to the book; this is more than adequate for the class and others using the book this term.)

SWI Prolog

We shall be doing programming in **Prolog** in this class. We shall use the **SWI-Prolog** implementation.

SWI-Prolog is installed and available on the PRISM computers (EECS's lab computers). And it is easily installed if you want to install it on your own computer for convenience. (And who wouldn't want to?! It is **Prolog**!) It is *free software* licenced under the GNU Lesser General Public License. By the way, "[t]he name 'SWI' is derived from 'Sociaal-Wetenschappelijke Informatica' ('Social Science Informatics')…" [*SWI-Prolog @* **Wikipedia**]

One interacts with **Prolog** via the *interpreter* within a **Prolog** interactive "shell". **SWI-Prolog**'s shell is called `swipl`. One enters it at the command line. That is,

```
% swipl
```

Once in the shell, well … there, you "talk Prolog"! To exit the shell, one types

```
?- halt.
```

at the top-level prompt. Or `<CTRL>-D` at the prompt.

Evaluation Components (Assignments, Tests, and Exams)

Evaluation Component	Key	Weight	When
Assignments	**A1(Q)/A2(Q)**	2*15%=30%	*as posted*
Midterm Test	**MT**	30%	June 23 (Mon)
Final Exam	**FE**	40%	August (TBD)

There will be *two* assignment sets (A1 and A2) through the semester.

Each assignment set may contain different parts and involve a mix of questions to be answered and programming. Answers should be typeset and not handwritten. Some of the questions may be offered in the form of quizzes. All your assignments must be submitted on eClass within the allowed period. Other means of submission are not acceptable.

Late submissions of the assignments are not allowed. However, students are welcome to use the Special Request form on eClass for submitting an appropriate special consideration requests when circumstances beyond their control interfere with the timely delivery of their coursework.

And there is a Midterm Test (MT) and a comprehensive Final Exam (FE).

If a student misses the Midterm Exam for a serious reason beyond their control, the student is welcome to submit the appropriate Special Request form on eClass.

If a student misses the Final Exam they will have to apply for Deferred Standing following the University regulations (the Special Request form on eClass is not applicable in such cases).

Schedule

This is the *tentative* schedule for the class. Topics may shift slightly depending on our progress and our interests, so this schedule may be updated along the way.

Week	Topics	Readings
1	Administrative matters. Introduction to AI (Why AI?). Intelligent Agents.	**RN**: Ch. 1 & 2
	I. Knowledge Representation & Reasoning	
2	Knowledge Representation and First-order Logic (FoL). Syntax and Semantics.	**RN**: Ch.8, review Ch.7
3	Introduction to Prolog. Core Concepts, Notations, Syntax, Backward Chaining.	**CM**: Ch.1, 2, 3(§1–3), 8
4	Logic Programming. Inference/Search as Computation. Inference in FoL. Unification, Resolution, Reasoning.	**RN**: Ch.9(§1, 2, 5)
5	Prolog: Control Flow, Backtracking, Negation.	**CM**: Ch.3(§4–8) , 4, 10
6	Prolog: Second-order Features, Tail Recursion.	**CM**: Ch.6
Reading Week (not counted)		
	II. Search & Planning	
7	**Midterm Test** (in the lecture room at lecture time) Introduction to Search.	**RN**: Ch.3(§1–3)
8	Undirected/Uninformed Search.	**RN**: Ch.3(§4)
9	Directed/Informed Search.	**RN**: Ch.3(§5, 6), 4(§1)
10	Constraint Satisfaction Problems. Backtracking Search.	**RN**: Ch.6
11	Planning as Heuristic Search.	**RN**: Ch.11(§1, 2, 4)
	III. Uncertain Knowledge and Reasoning	
12	Uncertain/Probabilistic Reasoning and Naïve Bayes. Review and onward.	**RN**: Ch.12(§1–3)

E-mail Policy

On e-mail from students with questions regarding course materials.

- I will not necessarily answer these e-mails by e-mail reply. My time is spent more productively for the class's sake in different ways. For pertinent questions on the materials that students send me by e-mail — or for questions that many people seem to be having — I will try to address them in class.

  Of course, if you have found I have not addressed your questions later adequately in class or in additional materials, send a follow-up email and I will get back to you.

  If you have an urgent question I have not addressed, the best would be to see me during my office hours.

On e-mail from students with issues regarding administrative matters.

- For administrative requests (e.g., "I cannot make the test,"), e-mail is fine, of course, and I will attempt to answer you directly.

Please include "`[EECS-3401]`" at the beginning of the *subject line* so we are able to sort it. Please be clear about who you are when e-mailing me or TAs — include your *name* and, *student#* and make sure that you email from your **my-york** account, so that we can know who you are.

Academic Integrity, Honesty, & Plagiarism

The Department of Electrical Engineering & Computer Science's Academic Honesty Guidelines are in effect for this course, as indeed, they are for any EECS course.

Plagiarism is defined as taking the language, ideas, or thoughts of another, and representing them as your own. If you use someone else's ideas, cite them. If you use someone else's words, clearly mark them as a quotation. Note that plagiarism includes using another's computer programs or pieces of a program. All noted instances of plagiarism will be reported.

These policies are not intended to keep students from working with other students. One can learn much working with others, so this is to be encouraged. Should you encounter any situations for which you are uncertain whether the collaboration is permitted or not, please ask.