



Artificial Intelligence Programming in Prolog

Lecture 1: An Introduction

References

- Useful references:
 - Clocksin, W.F. and Mellish, C.S., [Programming in Prolog: Using the ISO Standard \(5th edition\)](#), 2003.
 - Bratko, I., [Prolog Programming for Artificial Intelligence \(3rd edition\)](#), 2001.
 - Sterling, L. and Shapiro, E., [The Art of Prolog \(Second edition\)](#), 1994.

What is AIPP?

- A comprehensive introduction to Prolog.
- Specific focus on Artificial Intelligence programming techniques:
 - Knowledge representation and manipulation,
 - Database construction and management,
 - State-space Search,
 - Planning,
 - Meta-programming,
 - Text parsing and Definite Clause Grammars.

What is Prolog?

- PROgrammation et Logique.
- Edinburgh syntax is the basis of ISO standard.
- High-level interactive language.
- Logic programming language.
 - Based on Horn Clauses
 - $(\text{parent}(X,Z) \wedge \text{ancestor}(Z,Y)) \supset \text{ancestor}(X,Y)$

What is Prolog? (2)

- Programming languages are of two kinds:
 - **Procedural** (BASIC, ForTran, C++, Pascal, Java);
 - **Declarative** (LISP, Prolog, ML).
- In procedural programming, we tell the computer **how** to solve a problem.
- In declarative programming, we tell the computer **what** problem we want to solved.
- (However, in Prolog, we are often forced to give clues as to the solution method).

What is Prolog used for?

- Good at
 - Grammars and Language processing,
 - Knowledge representation and reasoning,
 - Unification,
 - Pattern matching,
 - Planning and Search.
 - i.e. Prolog is good at Symbolic AI.
- Poor at:
 - Repetitive number crunching,
 - Representing complex data structures,
 - Input/Output (interfaces).

SWI-Prolog

- SWI-Prolog is a good, standard Prolog for Windows and Linux
- It's licensed under GPL, therefore free
- Downloadable from: <http://www.swi-prolog.org/>

Syllogisms

- “Prolog” is all about programming in logic.
- Aristotle described syllogisms 2300 years ago
- Sample syllogism:
 - Socrates is a man.
 - All men are mortal.
 - Therefore, Socrates is mortal.
- This is logic. Can Prolog do it?

Syllogisms in Prolog

Syllogism

Socrates is a man.

All men are mortal.

Is Socrates mortal?

Prolog

man(socrates).

mortal(X) :- man(X).

?- mortal(socrates).

Facts, Rules, and Queries

- Fact: Socrates is a man.
 - `man(socrates).`
- Rule: All men are mortal.
 - `mortal(X) :- man(X).`
- Query: Is Socrates mortal?
 - `mortal(socrates).`
- Queries have the same form as facts

Running Prolog I

- Create your "database" (program) in any editor
- Save it as *text only*, with a **.pl** extension
- Here's the complete program:

```
man(socrates).  
mortal(X) :- man(X).
```

Running Prolog II

- Prolog is *completely interactive*. Begin by
 - Double-clicking on your .pl file, or
 - Double-clicking on the Prolog application and consulting your file at the ?- prompt:
 - ?- consult('C:\\My Programs\\adv.pl').
- Then, ask your question at the prompt:
 - ?- mortal(socrates).
- Prolog responds:
 - Yes

Prolog is a theorem prover

- Prolog's "Yes" means "I can prove it" --
Prolog's "No" means "I can't prove it"
 - ?- mortal(plato).
No
- This is the closed world assumption: the Prolog program knows everything it needs to know
- Prolog supplies values for variables when it can
 - ?- mortal(X).
X = socrates

Basic Elements of Prolog

- Our program is a database of **facts** and **rules**.
- Some are always true (**facts**):
father(john, jim).
- Some are dependent on others being true (**rules**):
**parent(Person1, Person2) :-
 father(Person1, Person2).**
- To run a program, we ask questions about the database.

Prolog in English

Example Database:

John is the father of Jim.
Jane is the mother of Jim.
Jack is the father of John.

Person 1 is a parent of Person 2 **if**
 Person 1 is the father of Person 2 **or**
 Person 1 is the mother of Person 2.

Person 1 is a grandparent of Person 2 **if**
 some Person 3 is a parent of Person 2 **and**
 Person 1 is a parent of Person 3.

} Facts
}
} Rules

Example questions:

Who is Jim's father?
Is Jane the mother of Fred?
Is Jane the mother of Jim?
Does Jack have a grandchild?

Prolog in Prolog

Example Database:

John is the father of Jim.
Jane is the mother of Jim.
Jack is the father of John.

Person 1 is a parent of Person 2 **if**
 Person 1 is the father of Person 2 **or**
 Person 1 is the mother of Person 2.

Person 1 is a grandparent of Person 2 **if**
 some Person 3 is a parent of Person 2 **and**
 Person 1 is a parent of Person 3.

Example questions:

Who is Jim's father?
Is Jane the mother of Fred?
Is Jane the mother of Jim?
Does Jack have a grandchild?

Example Database:

father(john, jim).
mother(jane, jim).
father(jack, john).

parent(Person1, Person2) :-
 father(Person1, Person2).
parent(Person1, Person2) :-
 mother(Person1, Person2).

grandparent(Person1, Person2) :-
 parent(Person3, Person2),
 parent(Person1, Person3).

Example questions:

?- father(Who, jim).
?- mother(jane, fred).
?- mother(jane, jim).
?- grandparent(jack, _).

Using Prolog (2)

- If you edit your program file (e.g. to correct something), be sure to consult it again afterwards!
- To exit from Prolog, type

|?- halt.
or press
Control/D

- The Prolog comment characters:
 - Single line comments: **%**
% This is a comment
This not a comment, but an error
 - Multiple line comments: **/***
/* This is a multi-line comment
which must be closed with a */