

Artificial Intelligence Programming in Prolog

Lecture 1:
An Introduction

Contents

- Course Details
- References
- What is Prolog?
- Prolog basics
- Prolog Demo

References

- Useful references:
 - Learn Prolog Now [Website](#)
 - Bratko, I., [Prolog Programming for Artificial Intelligence \(3rd edition\)](#), 2001

Software

- SWI Prolog
<http://www.swi-prolog.org>

What is Introduction to AI?

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

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
 - $(\text{parent}(X,Z) \wedge \text{ancestor}(Z,Y)) \Rightarrow \text{ancestor}(X,Y)$

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 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).

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.

Lecture 1: An Introduction

9

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.

Example questions:

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

} Facts
} Rules

Lecture 1: An Introduction

10

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

1. First, write your program (away from computer!).
2. Then, type it into a file, with a **.pl** extension.
 - Any text editor will do, but **Notepad++** or **Sublime** is recommended.
3. Then, run:

SWI-Prolog

4. You will be presented with the Prolog prompt

?-

5. Then, 'consult' your file (omitting the .pl):

?- consult(yourfilename). or
?- [yourfilename]. or **['folder/filename'].**

6. Then you can ask questions of your database.

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

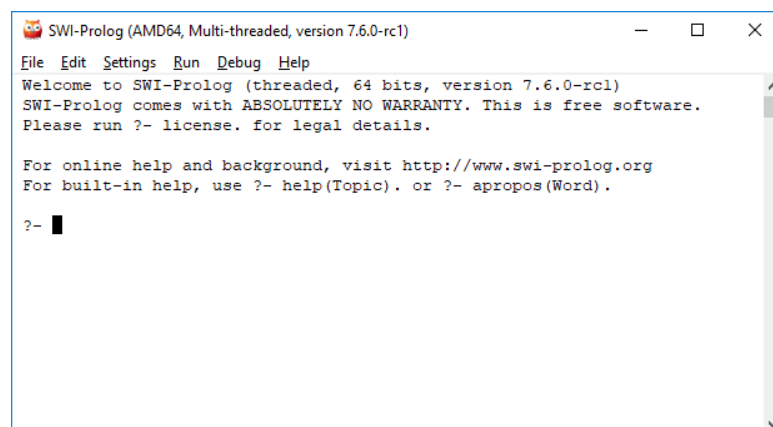
This not a comment, but an error

- Multiple line comments: **/***

/* This is a multi-line comment

which must be closed with a */

Prolog Demo



```
SWI-Prolog (AMD64, Multi-threaded, version 7.6.0-rc1)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.0-rc1)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
```