

# Artificial Intelligence Programming in Prolog

Lecture 1:  
An Introduction  
23/09/04

# Contents

- Course Details
- References
- Requirements and Exemptions
- What is AIPP?
- What is Prolog?
- Prolog basics
- Prolog Demo

# Introductions

- I am Tim Smith
  - [tim.smith@ed.ac.uk](mailto:tim.smith@ed.ac.uk)
  - Office 9, 2<sup>nd</sup> floor/left, 2 Buccleuch Place.
  - Office Hours 9am-12pm every Tuesday.
- You are....
  - Masters students
    - Informatics, AI, CogSci, SLP

# Course Details

- 19 lectures
- Mondays and Thursdays 16:10-17:00
  - A9/11, Ground Floor, Forrest Hill
- 2hr weekly lab tutorials
  - Computer Lab West, level 5, Appleton Tower
  - Wednesday 4-6pm or
  - Friday 3-5pm
    - Please record preference on sheet.

# Assessment

- Summer Examination 70%
- 2 Assignments
  - Assignment 1 = 10%
    - Available week 2. Due week 6
  - Assignment 2 = 20%
    - Available week 6. Due week 11.
- Weekly Practical exercises
  - Not assessed, but
  - Completion is compulsory.

# Course Materials

- **Course Notes (primary reference)**
  - Handed out in Monday's lecture
  - Contains:
    - Course details
    - Introduction to Prolog
    - Revision exercises
    - Weekly Practical exercises
  - Must bring to practical sessions
- **On-line Lecture slides**
  - <http://www.inf.ed.ac.uk/teaching/courses/aipp>

# References

- No course text book
- 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.

# Requirements & Exemptions

- AIPP is aimed at students with previous programming experience.
- If you have no, or little experience please take Introduction to Java Programming.
- If you are required to take AIPP and believe you may struggle with the programming speak to me.
- You may be exempt from AIPP if:
  - You have previously taken a Uni course in Prolog
  - You have used Prolog competently in industry.
- Speak to Specialism supervisor for exemption.



# Software

- Sicstus Prolog
- Installed on the Informatics DICE network
  - Type `sicstus` in a terminal window.
- Computer labs: 5<sup>th</sup> floor, Appleton Tower
- Free Windows version of sicstus available
  - Request a copy using the Informatics support form: <http://www.inf.ed.ac.uk/cgi-bin/support.cgi>
  - All prolog code must be tested on DICE version of sicstus before submission.

# 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 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.

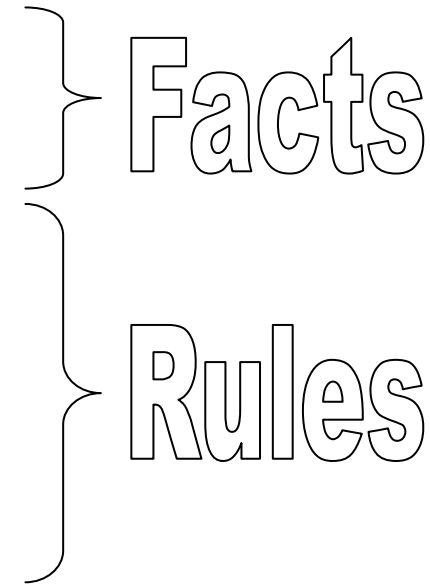
# Prolog in English

## 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?

# 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 **Emacs** is recommended.
3. Then, type:
 

```
sicstus
```
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. The entire content of your file is then stored in the memory of the Prolog interpreter.
  - You can see what is consulted by typing `| ?- listing.`
7. 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

```

SICStus 3.10.1 (x86-win32-nt-4): Fri Apr 11 23:08:29 WEDT 2003
File Edit Flags Settings Help
SICStus 3.10.1 (x86-win32-nt-4): Fri Apr 11 23:08:29 WEDT 2003
Licensed to dai.ed.ac.uk
| ?- ['imdb/actors_popular1'].
% consulting c:/program files/sicstus prolog 3.10.1/bin/imdb/actors_popular1.pl...
.
% consulted c:/program files/sicstus prolog 3.10.1/bin/imdb/actors_popular1.pl in
module user. 24408 msec 33949296 bytes
yes
| ?- actor('Kevin Bacon',Film,Date).
Date = 1994,
Film = 'Air Up There, The' ? ;
Date = 1978,
Film = 'Animal House' ? ;
Date = 1995,
Film = 'Apollo 13' ?
yes
| ?- actor('Kevin Bacon',Film,Date),actor('Dustin Hoffman',Film,Date).
Date = 1996,
Film = 'Sleepers' ? ;
no
| ?- █

```