Prolog (Programming in Logic)

Dr Ian Lewis, Professor Andrew Rice, Lent 2022

Hello and welcome to the Prolog course! You've already spent time learning where the commas and asterisks go in a variety of imperative programming languages (e.g. Java, Javascript, pretty much every other commonly used language). With a bit of luck you've also learned that using a declarative language based on the rigorous foundations of the *functional calculus* (e.g. ML) is still highly capable but provides greater opportunities to reason with clarity about the expected outcome of the execution. Prolog, based upon the *relational calculus*, presents another declarative paradigm for program execution particularly suited to particular problems (e.g. those involving searching through linked relations seeking a valid solution for a query).

I've included a few brief notes here to explain how the course works. I can be contacted at ijl20@cam.ac.uk if you have any questions.

The department webpage for this course is:
https://www.cl.cam.ac.uk/teaching/2021/Prolog

The videos and the PDF's of the slide decks are accessible via Moodle:
https://www.vle.cam.ac.uk/course/view.php?id=145061&sectionid=2027991

Note the Moodle videos include pauses with helpful 'checklist' questions, but as a 'backup' the same videos (without the checklist questions) is available via YouTube:
https://www.youtube.com/playlist?list=PLFgheI-d1nD47H06F236cp5cFEQE0o8nx

Course structure

This course consists of 8 timetabled 'lectures' starting on the 28th Feb 2022. The lectures have been designed with an 'inverted' structure i.e. videos are provided explaining the core content, and an in-person lecture (hopefully, Covid-19 permitting) is scheduled to review that material and clarify any subtleties or implications.

With reference to the materials list on the Moodle course page:
https://www.vle.cam.ac.uk/course/view.php?id=145061&sectionid=2027991
Overall *per-lecture* on the Prolog course, we have the following. Please note the optional elements are unrelated to the assessment or marking of the course:

- ➢ One or more short (10..20min) videos explaining the core material (required). These videos may contain embedded multiple-choice questions to check your understanding (optional)
- ➢ One or more Quiz to be viewed anytime after viewing the video to check your understanding (optional)
- ➢ The PDF slide deck (and possibly a video) of the in-person lecture (optional).

On that Moodle course page linked above you will find links to videos for all the content with the particular lecture slot with which they are best associated. If preferred you can also review this content at your own pace although that may then be out-of-sync with the in-person lectures.

Supervisions

There are two supervisions associated with the course and the "Recommended supervision work" is included on the Moodle course page linked above. Your supervisor might choose to set you something different.

Practical Work

For most of you, Prolog will be a new programming language using a new programming paradigm. The absolute #1 mistake to avoid early on in the course is to believe *all* programming languages are imperative and hence spend time trying to work out how to implement corresponding simple imperative features in your Prolog programs.

Familiarity with Prolog is most easily attained by writing a few simple programs while avoiding the #1 mistake listed above. SWI Prolog (https://www.swi-prolog.org/) is a mature implementation and I recommend you install that and experiment with it as early as is practicable. The 'hello world' of Prolog is to define a few relations of who is the child of who, and then create a rule that declares the meaning of a 'grandchild'.

Keep in mind that Prolog is designed for searching algorithms over data stored in the form of relations, and to do that does not require much of the real-world API support normal in common application languages. So if you find yourself trying to interface Prolog to the latest Windows .Net API or read data from a network socket, stop.

Contacting me

The Moodle course page has a help forum I will keep an eye on - that is particularly suited to queries related to the Prolog language and execution model (given some example) which will typically benefit others reading the forum.

You can also contact me at ijl20@cam.ac.uk if you need a friendly point in the right direction.