Introduction to
Prolog

Mihir Patil

Sushma Devaramani

Introduction

There are primarily two computer languages used in artificial intelligence work

- List Processing abbreviated as LISP, looks klutzy but it is based upon the lambda calculus and works quite well for computation associated with artificial intelligence.
- PROLOG has a lesser range of application when compared to LISP but allows for better formulation of the task. It is particularly good for solving problems regarding relationships.

A PROLOG program consists of :

- **Declaration of the facts of the relations involved.**
- **Declaration of rules concerning relations.**
- **Formulation of questions to be answered.**

Specifying Relationships

- In PROLOG they are defined in a functional form with the name of the relation first and the object or objects involved in the relation being enclosed within parentheses.

- Eg: aunt(jessica,liam). ,where aunt is known as the 'predicate' and jessica and liam are known as the 'atoms'.
- The conventions of PROLOG are:
  all names of relations and objects are in lower case letters.
  the objects are separated by commas.
  the relationships end with a full stop.

Running a PROLOG program

A prolog program is run by typing questions in the form:

- ?- aunt(jessica, liam). , to which the computer would respond: Yes.

- If the PROLOG program responds with a No to a question it does not mean that the statement involved in the question is false. It only means that it cannot be proved true with the given data.
- More sophisticated questions can also be asked using variables, variables are declared with a capital letter. Eg : ?- aunt(jessica, N). the computer would answer N=liam. At this point if we simply press return the computer stops searching for further matches. Instead if we use the semicolon ';' and then *return* the computer will search for any more objects for which jessica is an aunt.

Conjoining Questions

- We can conjoin two questions with a comma in between. The ',' corresponds to the logical 'and'. The comma is used to know if the answers to two questions correspond.
- Eg: ?- reads(jane, X), reads(jake, X).

Composing Relationships, using the 'if' condition

- Combinations of relations can be created by special PROLOG operations.

- grandparent(X,Z) :- parent(X,Y), parent(Y,Z). , the ':-' denotes 'if'.

Symbolic Computation

- Symbolic differentiation provides a good example of symbolic computation.
- The set of differentiation rules in PROLOG format are given below:

```
deriv(C,X,0) :- constant(C).
deriv(X,X,1) :- !.
deriv(-F,X,-G) :- deriv(F,X,G).
deriv(F+G,X,H+I) :- deriv(F,X,H), deriv(G,X,I).
deriv(F*G,X,H*G+F*I) :- deriv(F,X,H), deriv(G,X,I).
deriv(F^C,X,c*F^(C-1)*G) :- const(C), deriv(F,X,G).
deriv(F/G,X,H/G - (F/G^2)*I) :- deriv(F,X,H), deriv(G,X,I).
deriv(log(F),X,H*(F^(-1)) :- deriv(F,X,H).
```

- The rule that deriv(X,X,1) is always true, the meaning of ":- !." is "if anything."
- Since the output of the differentiation program is not simplified, it needs to run in conjunction with a simplification program to get a simplified output.

Citations:

http://www.sjsu.edu/faculty/watkins/prolog.htm