

ITEC 198 - Prolog Programming

Dr. Maung M. Htay

Department of Information Technology

1/10/2005

Text

Prolog Programming for Artificial Intelligence

by Ivan Bratko

Third Edition, Addison Wesley

References

- ◆ Artificial Intelligence by George F Luger, Fourth Edition, Addison-Wesley
- ◆ Prolog Help/References
<http://www.geocities.com/saviranid/>
- ◆ Roman Barták, 1997 page
<http://ktiml.mff.cuni.cz/~bartak/prolog.old/intro.html>

What is Prolog?

- ◆ **Programming in Logic**
- ◆ Developed in early 1970, by Robert Kowalski, Maarten van Emden, and David D.H. Warren at Edinburgh, U.K., Alain Colmerauer at Marseilles
- ◆ For symbolic, non-numeric computation
- ◆ Suited for solving problems that involve objects and relations between objects

Language Design

- ◆ centered around a small set of basic mechanisms
- ◆ including pattern matching, tree-based data structuring and automatic backtracking

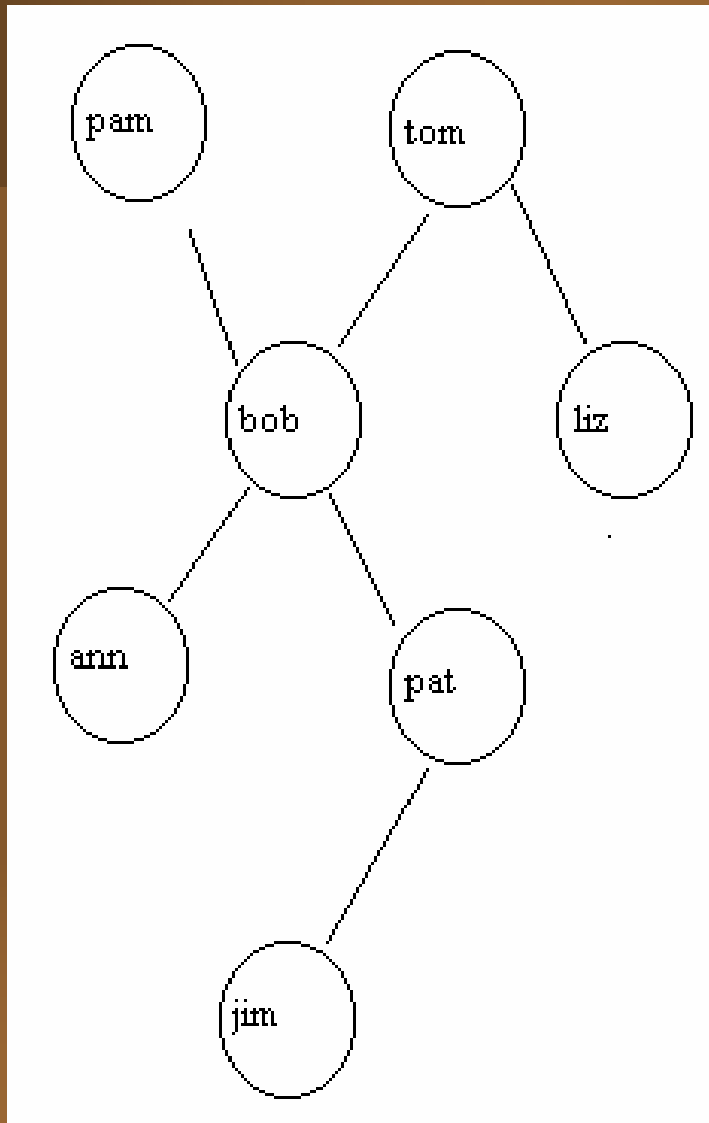
Chapter 1: Introduction to Prolog

- ◆ Defining relations by facts
- ◆ Defining relations by rules
- ◆ Recursive rules
- ◆ How Prolog answers questions
- ◆ Declarative and procedural meaning of programs

1.1 Defining relations by facts

- ◆ The fact that “Tom is a parent of Bob” can be written in Prolog as:
- ◆ `parent(tom,bob).`
- ◆ `parent` is the relation
- ◆ `tom` and `bob` are its arguments

A Family Tree



Prolog Program for the previous family tree

```
parent(pam,bob).  
parent(tom,bob).  
parent(tom,liz).  
parent(bob,pat).  
parent(bob,ann).  
parent(pat,jim).
```

Clauses

- ◆ A **clause** declares one fact about a relation

For example,

- ◆ `parent(tom,bob)` is a particular **instance** of the parent relation
- ◆ an instance is also called a **relationship**
- ◆ a **relation** is defined as the set of all its instances

Question to Prolog

For example,

- ◆ Is Bob a parent of Pat?

In Prolog,

- ◆ `?- parent(bob,pat).`

Prolog will answer:

- ◆ `yes`

More questions

A further query can be:

- ◆ ?- parent(liz,pat).

Prolog answers:

- ◆ no

More questions continue ---

Who is Liz's parent?

◆ ?- parent(X,liz).

So the answer is:

◆ X = tom

More questions continue ---

Who are Bob's children?

- ◆ ?- parent(bob,X).

The first answer is:

- ◆ X = ann

The another answer follows:

- ◆ X = pat

Broader Questions --

Who is a parent of whom?

In other words,

- ◆ Find X and Y such that X is a parent of Y.

In Prolog,

- ◆ `?- parent(X,Y).`

Broader Questions Continue ---

The answers are output as:

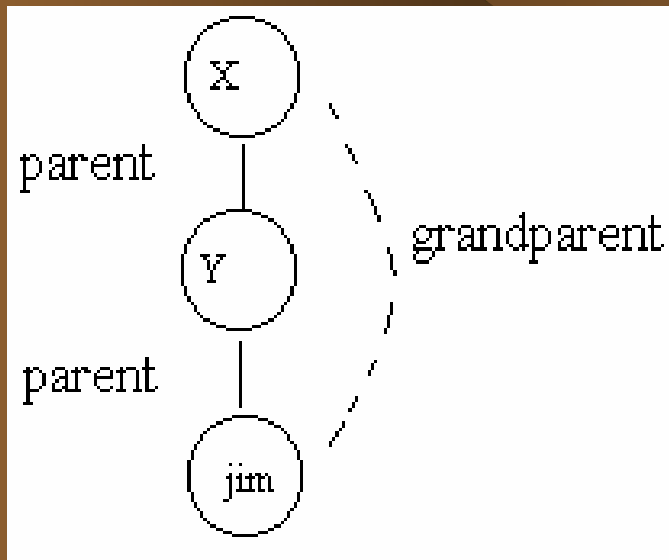
- ◆ X = pam
- ◆ Y = bob;

- ◆ X = tom
- ◆ Y = bob;

- ◆ X = tom
- ◆ Y = liz;
- ◆ ...

Composed Query in Prolog

- ◆ Who is a grandparent of Jim?



Composed Query in Prolog Continue --

To find a grandparent, we need two steps:

- ◆ Who is a parent of jim? Assume that there is some Y.
- ◆ Who is a parent of Y? Assume that there is some X.

In Prolog,

- ◆ `?- parent(Y, jim), parent(X,Y).`

Composed Query in Prolog Continue --

Who are Tom's grandchildren?

- ◆ ?- parent(tom,X), parent(X,Y).

Do Ann and Pat have a common parent?

- ◆ ?- parent(X,ann), parent(X,pat).

Important Points

- ◆ Easy to define a relation, by stating the n-tuples of objects that satisfy the relation such as **parent**
- ◆ Easy to query the Prolog system about relations defined in the program
- ◆ A Prolog program consists of **clauses**. Each clause terminates with a full stop
- ◆ Arguments of relations can be concrete objects, or constants (such as tom and ann), or general objects such as X and Y

Important Points Continue ---

- ◆ Concrete objects or constants are called **atoms** and general objects are called **variables**
- ◆ Questions to the system consist of one or more **goals** that are to be satisfied in the program such as: **?- parent(X,ann), parent(X,pat).**
- ◆ Answer can be positive (if satisfiable) or negative (if unsatisfiable)
- ◆ If several answers satisfy the question then Prolog will find as many of them as desired by the user

1.2 Defining relations by rules

More relations

Unary relations

- ◆ female(pam).
- ◆ male(tom).
- ◆ male(bob).
- ◆ female(liz).
- ◆ ...

Binary relations

- sex(pam,feminine).
- sex(tom,masculine).
- sex(bob,masculine).
- sex(liz,feminine).
- ...

Unary relations are simple yes / no properties of objects.

More Relations

Example

- ◆ to define a relation **offspring** as the inverse of the **parent** relation as a fact
- ◆ **offspring(liz, tom)** is inverse of **parent(tom, liz)**

It is understood as

Liz is an offspring of Tom if Tom is a parent of Liz.

In general, we can say that

Y is an offspring of X if X is a parent of Y.

Relations are Defined Elegantly

- ◆ to define **offspring** relation using already defined **parent** relation

For all X and Y,

Y is an offspring of X if X is a parent of Y.

In Prolog,

- ◆ `offspring(Y,X) :- parent(X,Y).`

What is Rules?

For all X and Y,
if X is a parent of Y then
Y is an offspring of X

In Prolog,

◆ `offspring(Y,X) :- parent(X,Y).`
is called a Rule.

Difference between facts and rules

- ◆ A fact like `parent(tom,liz)` is something always, unconditionally true.
- ◆ On the other hand, rules specify things that are true if some condition is satisfied.

Rules have

- ◆ body, a condition part (the right-hand side of the rule) and
- ◆ head, a conclusion part (the left-hand side of the rule)

The format is

offspring(Y,X) :- parent(X,Y).

head

body

More Rules

To define **mother** relation by rule

For all X and Y,

X is the mother of Y if

X is a parent of Y and

X is a female.

In Prolog,

◆ **mother(X,Y) :- parent(X,Y), female(X).**

More Rules Continue ---

To define **grandparent** relation by rule

For all X and Y,
X is a grandparent of Y if
X is a parent of Z and
Z is a parent of Y.

In Prolog,

- ◆ **grandparent(X,Y) :-**
- ◆ parent(X,Z),
- ◆ parent(Z,Y).

More Rules Continue ---

How do we define **sister** relation?

For all X and Y,

X is a sister of Y if

both X and Y have the same parent, and

X is a female.

In Prolog,

- ◆ **sister(X,Y) :-**
- ◆ parent(Z,X),
- ◆ parent(Z,Y),
- ◆ female(X).

Question to prolog

Who is pat's sister?

In Prolog,

- ◆ `?- sister(X,pat).`

The answer to the previous program

- ◆ `X = ann`
- ◆ `X = pat`

- ◆ We need to modify the program since pat is a sister of herself

Some Important Points, So Far

- ◆ Prolog program can be added new clauses
- ◆ Clauses are of three types: **facts**, **rules**, and **questions**
- ◆ Facts declare things that are always, unconditionally true
- ◆ Rules declare things that are true depending on a given condition
- ◆ Questions are to be asked by the user

More Important Points

- ◆ Clauses consists of head and body
- ◆ Body is a list of goals separated by commas
- ◆ Facts have a head and the empty body
- ◆ Questions only have the body
- ◆ Rules have the head and the non-empty body
- ◆ A variable can be substituted by another object, that is called, variable is **instantiated**