Introduction to Prolog

York University CSE 3401

Vida Movahedi

Overview

- Introduction & Preliminaries
- Syntax
  - Characters
  - Constants
  - Variables
  - Operators
- Arithmetic

[ref.: Chapter 2- Clocksin]

A sample program

```
student(john, 3401):-.
student(mary, 3401):-.
study_hard(john):-.
```

Facts

A rule

```
pass_3401(X):- student(X, 3401), study_hard(X).
```

```
:-pass_3401(john).
```

A goal

In Prolog

```
consult(user).
    student(john, 3401).
    student(mary, 3401).
    study_hard(john).

    pass_3401(X):- student(X, 3401), study_hard(X).

Ctrl+D
```

```
:-pass_3401(john).
true.
:- pass_3401(mary).
false.
```

Introductory notes

- Predicates shown by a name starting with lower case letters.
- Arguments are written in parentheses, separated by commas.
- A dot (period) comes at the end.
- The predicate name and the number of arguments is decided by the user.
- Queries are answered by matching with the knowledge given (facts and rules).
- A **false** means 'not enough knowledge to prove it' (does not mean it is actually false, maybe not enough knowledge given to Prolog).

In Prolog

```
consult(user).

student(john, 3401).
student(mary, 3401).
study_hard(john).

pass_3401(X):- student(X, 3401), study_hard(X).

Ctrl+D
```

A relationship defined by a predicate

An object constant

An object variable

Facts

A rule

```
:-pass_3401(john).
true.
:- pass_3401(mary).
false.
```

A satisfiable goal

An unsatisfiable goal or 'not able to prove satisfiable'

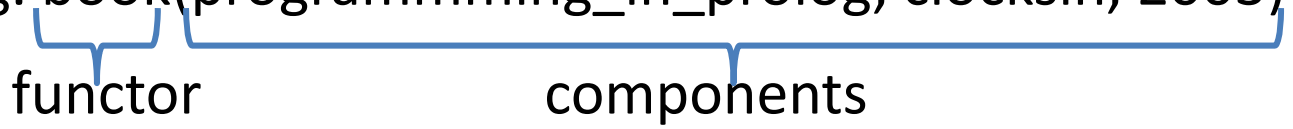
Goals

Syntax

- Characters
  - Upper case A, B, ..., Z
  - Lower case a, b, ..., z
  - Digits 0, 1, ...,9
  - Sign characters +, -, ...$, &
- Constants
  - Two kinds: Atoms and Numbers
  - Numbers, such as 25, 9.99, -30, 7.08e-15
  - Some symbols used for atoms are made up of letters and digits, must normally begin with a lower case, e.g. mary, john, likes, ...
  - If enclosed in single quotes, may have any character, e.g. ‘George-Smith’, ‘23g’, ...
  - Underline symbol ‘_’ may be in the middle
  - Some are made from signs only

Syntax (cont.)

- Variables
  - Their names begin with a capital letter or '_'
  - The anonymous variable: _ (with a peculiar characteristic!)
- Structures or compound terms
  - A single object consisting of a collection of other objects, called components
  - Defined as a functor and its components
  - e.g. book(programmming_in_prolog, clocksin, 2003)



  - Can use anonymous variable when not enough info e.g. book(book123,_,_)

Syntax (cont.)

- Operators
  - To make some functors easier to use, e.g. instead of +(3,4) we can write 3+4 (Important: it is not the same as 7)
  - Position
    - prefix, infix, or postfix, e.g. +(3,4), 2*5, 7!
  - Precedence
    - An integer associated with each operator, the closer to 1, the higher the precedence
    - e.g. multiplication has a higher precedence than addition, a-b/c is –(a, /(b,c))
  - Associativity
    - Left or right
    - All arithmetic operators left associative
    - e.g. 8/4/4 is (8/4)/4

Arithmetic

- X + Y, X - Y, X * Y, X / Y
- X // Y, X mod Y (integer quotient and remainder)
- X =:= Y, X =\= Y (same numbers, different numbers)
- X < Y, X > Y, X =< Y, X >= Y (comparisons)
- is: an infix operator, evaluating the right-hand argument, matched with left-hand argument

Example

```
pop(china, 800).
area(china, 4).
density(X,Y):-
    pop(X,P),
    area(X,A),
    Y is P/A.
```

The population density of a country X is Y, if:
The population of X is P, and
The area of X is A, and
Y is calculated by dividing P by A.

```
:- density(china, X).
X=200.
```