Prolog - Logic Programming

Mark Boady

August 8, 2013

Prolog

- A language for Logic Programming
- Database
  - Rules and Facts are stored in files and compiled before execution
- Queries
  - Queries are given to the interpreter
  - Variables: Uppercase or underscore before lowercase
  - Predicates/Rules: lower case names
- GNU Prolog is installed on the server
  - http://www.gprolog.org/
- To start: prolog
- To quit: Ctrl-D or halt.
- All Prolog statements end with periods

Using Prolog

- The append function is built into prolog.
- Everything is prolog evaluates to true or false.
- append(X,Y,L)
  - The function is true is L is the list resulting from appending X to Y.
- We can query a true/false case
  - append([1,2,3],[4,5,6],[1,2,3,4,5,6]).
  - append([1,2,3],[4,5,6],[]).
- We can put a variable in a query and find the values that make it true.
  - append([1,2,3],[4,5,6],X).
  - append(X,Y,[1,2,3,4,5,6]).
- To try for a new answer semi-colon
- To print all answers a
- To accept current answer hit enter

Custom Rules

- We can give prolog a file with custom rules
- We can't override the built in append.
- File Contents for append1.pl:

```
append1([], Y, Y).
append1([ A | B], Y, [A|W]) :- append1(B, Y, W).
```

- We can load the file by entering [append1].
- Alternatively, you can enter directly into prolog
  - Type [user].
  - Enter your facts and rules
  - Type Ctrl-D (End of File)

List Functions

- The vertical bar splits a list into the head and tail
  - [A | B] means that A is the head of the list and B is the tail
- We can use this to define our common list methods

```
car([A|B],A).
cdr([A|B],B).
cons(A,X,[A|X]).
```

- Note that we can't return the new list with cons. We need an addition input with the merged list

```
| ?- cdr(X,[2,3,4]).
X = [_,2,3,4]
yes
```

Trace

```
| ?- trace.
The debugger will first creep -- showing everything
yes
{trace}
| ?- append1([1,2,3],[4,5,6],X).
1    1  Call: append1([1,2,3],[4,5,6],_28) ?
2    2  Call: append1([2,3],[4,5,6],_61) ?
3    3  Call: append1([3],[4,5,6],_88) ?
4    4  Call: append1([],[4,5,6],_115) ?
4    4  Exit: append1([],[4,5,6],[4,5,6]) ?
3    3  Exit: append1([3],[4,5,6],[3,4,5,6]) ?
2    2  Exit: append1([2,3],[4,5,6],[2,3,4,5,6]) ?
1    1  Exit: append1([1,2,3],[4,5,6],[1,2,3,4,5,6]) ?
yes
{trace}
| ?- notrace.
The debugger is switched off
```

Cut

- The cut command tells prolog when to not try different alternativs
- An Example without cut:

```
a(X, Y) :- b(X), c(Y).
b(1).  b(2).  b(3).
c(1).  c(2).  c(3).
```

- We can query | ?- a(Q,R).
- Prolog gives us 9 possible answers

Cut

- Now, we can try the same command with a cut (!) added.
- An Example with cut:

```
a(X, Y) :- b(X), !, c(Y).
b(1).  b(2).  b(3).
c(1).  c(2).  c(3).
```

- We can query | ?- a(Q,R).
- Prolog only gives 3 answers
- The cut says, once a valid answer for b(X) is found never try again
  - Only b(1) is attempted
- Cut is important when optimizing code.

Infinite Answers

- Prolog will try to fine all possible answers, this can cause an infinite loop
- Member is built in to prolog

```
member1(X,[X|T]).
member1(X,[Y|T]):-member1(X,T).
```

- There are an infinite number of answers to the query member1(2,X).

```
X = [2|_] ? ;
X = [_,2|_] ? ;
X = [_,_,2|_] ? ;
```

- [2 | _] means any list with 2 as the first element is valid.
  - Underscore means anything
  - The tail of the list can be null or any length
- [_,2 | _] means a list with 2 as the second element

Lab

- Experiment with three prolog functions
- gcd - an alternative way to define the gcd
- last.pl - Find last element
- merge.pl - A mergesort algorithm