# Data Structures in Prolog - Function Symbols and Lists

Theoretically speaking, Prolog does not need any special data structures. Function symbols can serve the purpose of "structuring data". It is entirely up to the user to choose how something is represented.

Example - representing lists using the symbol cons.

append(nil, X,X).  
append(cons(A, L1), L2, cons(A, L3)) :- append(L1, L2, L3).

In this program, lists are represented as follows:

() by nil  
(a) by cons(a, nil)  
(a b) by cons(a, cons(b, nil)), etc.

Query:

?- append(cons(a1, cons(a2, nil)), cons(b1, nil), Q).  
  
Q = cons(a1,cons(a2,cons(b1,nil)))

It's perfectly fine to use any function name you like to represent lists:

append(e, X, X).  
append(c(A, L1), L2, c(A, L3)) :- append(L1, L2, L3).

In this definition, lists are represented as

() by e  
(a) by c(a, e)  
(a b) by c(a, c(b, e)), etc.  
  
?- append(c(a1, c(a2, e)), c(b1, e), Q).  
  
Q = c(a1,c(a2,c(b1,e)))

For convenience, efficiency and standardisation/code reuse, Prolog uses a builtin list structure聽**with square brackets**.

## Builtin List Structure

Lists are represented as

[] : empty list  
[a, b, c] : a list of three elements  
[F|R] : a pattern which will match a list of at least one element F,  
 with the rest of the list matching R  
  
E.g. [A|L] and [a,b,c] are **unifiable** setting A = a, and L = [b,c]  
  
[a,b|L] : a list of elements a, b and the rest is L  
  
[[a,b],c|L] : a list can be nested  
  
Note that [a|[]] = [a] , as in Lisp

Using this builtin structure, we can define append:

append([], X, X).  
append([A| L1], L2, [A|L3]) :- append(L1, L2, L3).

For testing purposes, it's convenient to write down the queries that you want to test, e.g.

p(W) :- append([a1, a2], [b1], W).  
q(X,Y) :- append(X,Y, [a1, a2]).

so that you don't have to type, e.g. ?- append([a1, a2], [b1], W) every time you want to run it.

% Now, execute queries  
  
| ?- p(A).  
  
A = [a1,a2,b1] ?   
  
yes  
| ?- q(A,B).  
  
A = [],  
B = [a1,a2] ? ;  
  
A = [a1],  
B = [a2] ? ;  
  
A = [a1,a2],  
B = [] ? ;  
  
no

Last modified: Monday, 29 February 2016, 8:43 PM