Examples of Prolog - Append

% myappend(X, Y, Z) is true if Z is the list of elements
% of X follows by elements of Y

myappend([], X, X).
myappend([H|T], Y, [H|Z]):-
    myappend(T, Y, Z).
Example - Member

% mymember(X, Y) is true if X is a member of the list Y

mymember(X,[X|_]).
mymember(X,[Y|Z]):-
    Y \== X, mymember(X,Z).
Example - Length

% mylength(X, N) is true if N is the number of elements of X

mylength([], 0).
mylength([H|T], N1):-
    mylength(T,N), N1 is N+1.
Example - Diff

% mydiff(X, Y, Z) is true if Z is a list obtained by
% removing all instances of elements of Y from X

mydiff(X, [], X).
mydiff(X, [Y|L], Z):-  \+ mymember(Y, X), mydiff(X, L, Z).
mydiff(X, [Y|L], Z):-  mymember(Y, X), myremove(X, Y, X1),
                      mydiff(X1, L, Z).

% myremove(X, Y, Z) is true if Z is a list obtained by removing all
% instances of Y from X note the difference between
% 'myremove(X, Y, Z)' and % 'mydiff(X, Y, Z)' -- one deals with ONE
% element and % the other deals with list

myremove([],_,[]).
myremove([H|T], H, T1):-  myremove(T, H, T1).
myremove([H|T], H1, [H|T1]):-  H \== H1, myremove(T, H1, T1).
Example - Shorter

% shorter(L1, L2) is true if L1 contains the same % number or fewer elements than L1

shorter([], []).
shorter([X|Y], [H|T]) :- shorter(Y, T).

newshorter(X, Y) :-
    mylength(X, N1),
    mylength(Y, N2),
    N1 =< N2.
Example – subset and samemembers

% subset(X, Y) is true if every member of X is
% a member of Y

subset([], X).
subset([H|T], Y):-
    mymember(H, Y),
    subset(T, Y).

% samemembers(X, Y) is true if X contains the same
% elements as Y

samemembers(X, Y):-
    subset(X, Y),
    subset(Y, X).
Example - Ancestor

% ancestor

father(tom, john).
father(tom, sandra).
father(john, mike).
father(john, elena).
mother(sandra, katy).
mother(elena, jerry).

ancestor(X, Y):- father(X, Y).
ancestor(X, Y):- mother(X, Y).
ancestor(X, Y):- ancestor(X, Z), ancestor(Z, Y).
Example – ‘Another’ Ancestor

% ancestor

father(tom, john).
father(tom, sandra).
father(john, mike).
father(john, elena).
mother(sandra, katy).
mother(elena, jerry).

newancestor(X, Y):- father(X, Y).
newancestor(X, Y):- mother(X, Y).
newancestor(X, Y):- father(X, Z), newancestor(Z, Y).
newancestor(X, Y):- mother(X, Z), newancestor(Z, Y).
Prolog - Problem?

- SLD resolution:
  - Left-to-right selection of atom
  - Top-down selection of clause
- Problem: sound but incomplete
- How to cope with the problem?
  - Writing recursion clauses: the basic case comes first;
  - No recursion within a clause: body should not contain an atom that can be resolved with the head;