

```

:- use_module(library(lists)). % subseq0.
% set/1 - Are the given elements one set?
set([]). % Empty set is trivially sorted.
% 'Sort' also removes duplicates, for set.
set(Elements) :- sort(Elements, Elements).

% union/3 - A set union of two sets.
union([], [], []). % Obviously.
union(A, [], A). union([], B, B).
union(LeftSet, RightSet, SetUnion) :-
    set(LeftSet), set(RightSet),
    % Append sets (which are lists).
    append(LeftSet, RightSet, Merge),
    % Sort the unordered 'Merge'd.
    sort(Merge, SetUnion).

% intersection/3 - Simple set intersection.
intersection([], [], []). % Obviously... Again...
intersection(_, [], []). intersection([], _, []). % Needed??!
intersection([L|LeftSet], [R|RightSet], [L|SetIntersection]) :-
    L == R, % Must exist in both LeftSet and the RightSet.
    set([L|LeftSet]), set([R|RightSet]), % Only on sets.
    intersection(LeftSet, RightSet, SetIntersection).
intersection([L|LeftSet], [R|RightSet], SetIntersection) :-
    L @> R, % There are no larger elements in R (invariant).
    set([L|LeftSet]), set([R|RightSet]), % Only on sets.
    intersection([L|LeftSet], RightSet, SetIntersection).
intersection([L|LeftSet], [R|RightSet], SetIntersection) :-
    L @< R, % There are no larger elements in L (invariant).
    set([L|LeftSet]), set([R|RightSet]), % Only on sets.
    intersection(LeftSet, [R|RightSet], SetIntersection).

% power_set/2 - Produces the power set of a predefined set.
power_set([], []). % Per the definition it has the zero set as well.
power_set(Set, PowerSet) :- set(Set), setof(X, subseq0(Set, X), PowerSet).
% Select a set of all possible subsequences 'Set' producing a 'PowerSet'.

% -----
% -----EXAMPLE QUERY-----
% -----
%
% Is [a, b, c] a valid set?
% -----
%
% ?- set([a, b, c]).
% yes
%
% Is [c, b, a] a valid set?
% -----
%
% ?- set([c, b, a]).
% no
%
% What is the union of [a, b, c] and [d, e, f]?
% -----
%
% ?- union([a, b, c], [d, e, f], S).
% S = [a,b,c,d,e,f] ? ;
% no
%
% What is the union of [a, b, c, d] and [d, e, f]?
% -----
%
% ?- union([a, b, c, d], [d, e, f], S).
% S = [a,b,c,d,e,f] ? ;
% no
%
% Can we do a union of [c, b, a] and [d, e, f]?
% -----
%
% ?- union([c, b, a], [d, e, f], S).
% no
%

```

```
% What is the intersection of [a, b, c] and [c, d, e]?
% -----
% ?- intersection([a, b, c], [c, d, e], S).
% S = [c] ? ;
% no
%
% What is the power set of the set [a, b, c]?
% -----
% ?- power_set([a, b, c], S).
% S = [[],[a],[a,b],[a,b,c],[a,c],[b],[b,c],[c]] ? ;
% no
```