Printed by rueher from ornithorynque.essi.fr


```
I)
?-use_module(library(lists)).

robot(As,N):-
        etat_i(E1),
        trans([E1],As,N).
trans([E|_],[],_):-
        etat_f(E).
trans([E1|El],[A|Al],N):-
        length(El,L),
        L < N,
        transfert(A,E1,E2),
        \+ member(E2,El),
        trans([E2,E1|El],Al,N).
transfert(t(X,Y,Z),E1,E2):-
        sur(X,Y,E1),
        libre(X,E1),
        libre(Z,E1),
        X \== Z ,
        transform(t(X,Y,Z),E1,E2).
etat_i([sur(c,1),sur(b,2),sur(nil,3),sur(nil,a),sur(nil,b),sur(a,c)]).
etat_f([sur(nil,1),sur(nil,2),sur(c,3),sur(nil,a),sur(a,b),sur(b,c)]).

libre(X,E):-
        member(sur(nil,X),E).
sur(X,Y,E):-
        member(sur(X,Y),E).
transform(t(X,Y,Z),Ei,Ef):-
        replace(sur(X,Y),sur(nil,Y),Ei,E1),
        replace(sur(nil,Z),sur(X,Z),E1,Ef).
replace(X,Y,[X|T],[Y|T]).
replace(X,Y,[Z|T],[Z|T1]):-
        replace(X,Y,T,T1).

/*
?- robot(A,5).

A = [t(a,c,b),t(c,1,3),t(a,b,1),t(b,2,c),t(a,1,b)]

*/

II) Interprete Prolog (cf. cours)

III)
1)
On suppose que Set est un ensemble (i.e.,des listes sans doublons). Comme pour
|Set|=N,  |Power_Of_Set|=2^N et qu'il a (2^N)! manieres d'ordonner sa
representation, on suppose que  Power_Of_Set est ordonne de la maniere
suivante:
a) Les ensembles les plus grands precedent les plus petits
   (Set est le premier, [] est le dernier)
b) Si deux ensembles ont la meme cardinalite, alors l'ordre des elements
   est utilise ([1] precede [2]).
2)
power_set(Set, Power_Of_Set) :-
    Power_Of_Set = [Set|_],
    power_set_case(Set, Power_Of_Set).

power_set_case([], [[]]).                              % Step 1.
power_set_case([Elt|Rest], Power_Of_Set) :-            % Step 2.
    power_set(Rest, Power_Of_Rest),                    % Step 2.1.
    power_set_augment(Power_Of_Rest, Elt, Power_Of_Set, Power_Of_Rest). % 2.2.

power_set_augment([], _, Power, Power).
power_set_augment([Subset|Subsets], Elt, [[Elt|Subset]|Power0], Power) :-
    power_set_augment(Subsets, Elt, Power0, Power).


/*
?- power_set(X, [[1,2],[1],[2],[]]).
X = [1, 2]
?- power_set(X, [[1,2],[2],[1],[]]).
no
?- power_set([1,2,3], X).
X = [[1, 2, 3], [1, 2], [1, 3], [1], [2, 3], [2], [3], []]
*/
```