Urban's Rule for Generating a List from other Lists

DESCRIPTION

$L_1, ..., L_n$ are given lists – of variable lengths. From each list L_i one element is chosen so that these elements make a new list N^x. All these new lists N^x together form a list LL of lists $N^1, ..., N^m$ such that any two new lists N^j and N^k are different.

```
cart([ ],_,[ ]):-!.
cart(_,[ ],[ ]):-!.
cart([A | ARest],BL,CL):-
    create_tuple(A,BL,TList),
    cart(ARest,BL,CInterim),
    append(TList,CInterim,CL).

create_tuple(_,[ ],[ ]).
create_tuple(A,[B | BRest],[[A,B] | TRest]):-
    create_tuple(A,BRest,TRest).

% the product of n lists is the empty list, if one of the input lists is empty.
% predicate is undefined, if by mistake the input list contains only one list.

n_cart(NList,[ ]):- member(NList,[ ]),!.
n_cart(NList, undef):- length(NList,1),!.
n_cart([AL,BL | Rest],CL):-          % take the first two lists and create
                                       the product of these two lists first %
    cart(AL,BL,C),            % now extend the initial tuples by processing
                                the remaining lists so that you get a list of
                                n-tuples which is the product %
    extend_cart_tuple([C | Rest],CL).

extend_cart_tuple([C | [ ]],C).
extend_cart_tuple([TupleList,Next | Rest],CL):-
    help_extend_cart_tuple(TupleList,Next,ETupleList),
    extend_cart_tuple([ETupleList | Rest],CL).

help_extend_cart_tuple([ ],_,[ ]).
help_extend_cart_tuple([T | TupleList],List,ExtTupel):-
    extend_tuple(T,List,EList),
    help_extend_cart_tuple(TupleList,List,IList),
    append(EList,IList,ExtTupel).

extend_tuple(_,[ ],[ ]).
extend_tuple(T,[E | Rest],[ExtT | ERest]) :-
    append(T,[E],ExtT),
    extend_tuple(T,Rest,ERest).
```

```
test(C):-
    cart([1,2],[3,4],C).

test0(C):-
    n_cart([[1],[3,4],[1,6,7]],C),
    assert(result(C)).

test(A,B,C):-
    n_cart([[1,2,3],[2],[4,5],[6,7,8],[9]],A),
    assert(result(A)),
    n_cart([[1,2],[3,4],[5,6],[7,8],[9,10],[11,12,13,14,15]],B),
    assert(result(B)),
    n_cart([[1,2],[ ],[5,6]],C),
    assert(result(C)).
```