Logic programming exercise (PROP HT2014)

Your task is to define a set of predicates for handling set operations. The following predicates are required:

union(+A,+B,−C): C is the union of A and B.

Example:
?- union([a,b,c], [b,c,d,e], C).
C = [a,b,c,d,e]

Note: There should be no duplicates in the resulting list. You may assume that the two first lists do not contain duplicates.

intersection(+A,+B,−C): C is the intersection of A and B.

Example:
?- intersection([a,b,c], [c,d,e,b], C).
C = [b,c]

difference(+A,+B,−C): C is the set of elements in A that are not present in B.

Example:
?- difference([b,a,c], [b,d,c,e], C).
C = [a]

subset(+A,+B): A is a subset of B.

Example:
?- subset([b,c], [c,d,b,e]).
yes

equal(+A,+B): the set A is equal to the set B.

Example:
?- equal([a,b,c], [c,a,b]).
yes

cartesian_product(+A,+B,?C): C is the set of all possible pairs that can be formed of the elements in A and B.

Example:
?- cartesian_product ([a,b,c], [1,2], C).
C = [(a,1), (a,2), (b,1), (b,2), (c,1), (c,2)]

Hint: for each element in the first list, you may use a predicate make_pairs(X,M,P) that produces a list of pairs in the following way: make_pairs(a, [1,2], P) results in P = [(a,1), (a,2)], after which the generated lists are appended.