

Programming Assignment #1 – Due Monday, 9 Feb.

Problem Statement:

For this programming assignment, you will gain some initial experience programming in Standard ML with its type inferencing and polymorphic type features (which can be a bit frustrating at first as you acquire proficiency in ML).

Simple Finite Set Theory

For this assignment, you are to define a set of functions for doing basic operations on sets of elements in the form of lists. The functions should operate over any type of list. For example: [1,2,3,4,5,6,7,8,9,10], ['a','b','c'], ["this","is","a","set","of","words"], and [] (the empty list). Note that a set can contain other sets. For example, the powerset of the set [1,2,3] is the set [[],[1],[2],[3],[1,2],[1,3],[2,3],[1,2,3]]. The set operations must include, but are not limited to, the following:

set – given a list of elements all of the same type T, the set operation removes duplicate elements and optionally orders the elements according to the order relation for elements of type T. For example, set ([2,2,1,3,0,9,1,3]) returns either [2,1,3,0,9] or optionally [0,1,2,3,9]. Ordering the elements simply improves the efficiency of some of the following operations.

empty – given a set, compute a predicate that decides whether or not the set is equal to the empty set (i.e., the empty list for that type of set).

cardinality – given a set S, compute the cardinality (i.e., size) of the set.

member – given an element x (which may be a set) and a set S (which may be a set of sets), compute the predicate that decides whether or not x is a member of S.

equality – given two sets, compute a predicate that decides whether the two sets are equal. Two sets are equal if they have the same elements, irrespective of the order of the elements.

union – given two sets, compute the union of the sets.

intersection – given two sets, compute the intersection of the sets.

product – given two sets, compute the cross product of the sets.

powerset – given a set S, compute the power set P(S).

subset – given two sets S1 and S2, compute a predicate that decides whether S1 is a subset of S2.

For the functions equality, member, union, intersection, product and subset, allow these operations to be used as either prefix functions or left associative infix operators defined using the SML **infixl** directive. The infix operators will have different names; for example `3 'elem' [1,2,3] => true` and `member (0, [1,2,3]) => false`.

Submission Requirements:

You are to hand in at the start of class on the due date a source code listing of your program with your name and email address on a cover sheet as well as in a comment line in the source files.

You are also to submit an electronic copy of your source program to the course TA via email.

Honor Policy:

This assignment is a individual learning opportunity that will be evaluated based on your ability to think independently, work through a problem in a logical manner and implement a software program on your own. You may however discuss verbally or via email the general nature of the conceptual problem to be solved with your classmates, the course TA or the course instructor, but you are to complete the actual programming for this assignment without resorting to help from any other person or other resources that are not authorized as part of this course. If in doubt, ask the course instructor. **You may not use the Internet to search for solutions to the problem.**