

Homework 5

CSC 4101, Fall 2011

Due: 1 December 2011

1. (10 pts)

Write set membership test and set intersection programs in each of the languages Scheme and ML. Each function will only be about two or three lines long.

As data structure for sets, we will use unordered lists. E.g., the Scheme list '(11 15 3) represents the set {3, 11, 15}. Each element should appear only once in the set.

Write Scheme functions `memv?` and `intersection` of the following types. The boolean function `memv?` takes an element and a list and returns true if and only if the element is in the list. Use the comparison `eqv?` for comparing individual elements. The function `intersection` takes two lists and returns a list representing the set intersection of the arguments. I.e., the result list contains all the elements that are in both argument lists.

Similarly, write two ML functions `member` and `intersection`. For comparing individual elements, you can use `=`.

You can test these programs, but for submitting the homework, it's good enough to have them on paper. The ML compiler is available as `/usr/bin/sml` on the classes server.

2. (10 pts)

Given the following ML declarations of a binary tree type with integers in the leaf nodes and of five functions operating on trees:

```
datatype tree = L of int                (* leaf with an integer *)
              | N of tree * tree       (* node with two subtrees *)

fun max (n, m) = if n > m then n else m

fun a (L i)      = 1
  | a (N (l, r)) = a l + a r

fun b (L i)      = i
  | b (N (l, r)) = b l + b r

fun c (L i)      = 0
  | c (N (l, r)) = 1 + c l + c r

fun d (L i)      = i
  | d (N (l, r)) = max (d l, d r)
```

```

fun e (L i)      = 0
  | e (N (l, r)) = 1 + max (e l, e r)

```

In each function definition, the first line is the case for a leaf node (i.e., the base case of the recursion), and the second line recursively calls the function for the left and right subtrees.

These functions compute:

- the height of the tree, and
- the number of leaves in the tree,
- the number of interior nodes in the tree,
- the sum of all the values in the leaves,
- the maximum value of any leaf node in the tree.

Associate the functions with their descriptions. Try it first without typing them in.

3. (10 pts)

Given the following Prolog program for set membership and set intersection:

```

mymember(E, [E | _]).
mymember(E, [_ | T]) :- mymember(E, T).

intersect([], _, []).
intersect([H | T], L2, [H | L3]) :-
    mymember(H, L2), intersect(T, L2, L3).
intersect([_ | T], L2, L3) :- intersect(T, L2, L3).

```

Show the steps a Prolog interpreter makes in finding the first solution of the query

```

intersect([1, 2], [2, 3], X).

```