Foundations of Computer Science – Supervision 4 Supplementary Sheet

Heidi Howard

November 7, 2014

Exercise 1:

This weeks warm-up exercise are as follows:

```
(* given a predicate p and binary tree, exists_depth returns
 true if p x is true for some tree element x and false
 otherwise, the tree is traversed in a depth-first manner *)
val exists_depth = fn: ('a -> bool) -> 'a tree -> bool

(* given a predicate p and binary tree, exists_breadth returns
 true if p x is true for some tree element x and false
 otherwise, the tree is traversed in a breadth-first manner
  and may use append*)
val exists_breadth = fn: ('a -> bool) -> 'a tree -> bool

datatype 'a queue = Queue of 'a list * 'a list

(* given a queue, rev reverses the order of the
        elements in the queue *)
val rev = fn: 'a queue -> 'a queue

(* given a univariate polynomial and a real, eval
        calculates the values of the polynomial *)
val eval = fn: (int * real) list -> real -> real

(* given a list, to_array returns an array containing
        the same elements *)
val to_array = fn: 'a list -> 'a array

(* given an array, from_array returns a list containing
        the same elements *)
val from_array = fn: 'a array -> 'a list

datatype 'a mlist = Nil
                  | Cons of 'a * 'a mlist ref

(* given a list, to_mlist returns a mutable list containing
        the same elements *)
```

```
val to_mlist = fn: 'a list -> 'a mlist

(* given an mutable list, from_mlist returns a list containing
 the same elements *)
val from_mlist = fn: 'a mlist -> 'a list
```