Homework Assignment #4

Due Monday, April 30, at the start of lecture. As always, turn in a typed hardcopy of your answers. You should not use any advanced functions that you find in the text or elsewhere to aid you in solving these problems; you should only use the functions that we’ve discussed in class.

1. For the version of association lists implemented as part of homework #3, implement a function alist_map: (''k*'v -> 'a) * (''k,'v) assoc_list -> 'a list that takes a mapping function f and an association list and returns the list of results of applying f to each key/value pair of the association list.

2. Use alist_map to implement a function team_names that extracts the list of all the team names from an association list of type Records from homework #3. Use an anonymous function as the argument to alist_map. Show how your function works on sample inputs.

The analogue for association lists of the list reduce function is

alist_reduce: (''k*'v*'a -> 'a) * 'a * (''k,'v) assoc_list -> 'a
which takes a reduction function f, an initial base value b, and an association list and returns the result of successively applying f to each key and value in the association list and result of previous reductions to produce a new reduction result, starting with b as the initial reduction result of the empty association list. It can be defined as follows:

fun alist_reduce(f, b, nil) = b
  | alist_reduce(f, b, (k,v)::rest) = f(k, v, alist_reduce(f, b, rest))

3. Use alist_reduce to define a function

best_record: Records -> {wins:int, losses:int} that computes the maximum team record, using better_record as the comparison function. Use an anonymous function as the argument to alist_reduce.

4. Write a function all_games_back: Records -> (string * real) list that produces for each team the number of games back that team is from the first-place team. The number of games back is computed as (the number of additional losses relative to the first-place team - the number of additional wins) / 2. Use best_record and either alist_map or alist_reduce, whichever is most appropriate, in your solution. Show how your function works on sample inputs. (Recall that to convert an int to a real you can use the function real: int -> real.)