Functional Programming
Exercises Week 4
(for November 8, 2013)

Numbers in parentheses refer to the 6th edition of the course notes.
Exercises marked with ⋆ are optional and can be ignored.
Please have your `.ml` files accessible (e.g. on `zid-gpl.uibk.ac.at`).

1. Read Chapter 4 of the lecture notes.

2. (Exercise 4.2) Compute (on paper) the Huffman encoding for the text:

```
DON'T PANIC
```

3. (Exercise 4.4) Consider the module `Huffman`. The function `table` (see Section 4.3.3[1]) performs a top to bottom construction of the code table, i.e., it builds the code when descending in the tree and when arriving at a leaf the code (for a single character) is already computed.

A bottom to top function first descends in the tree and then generates the codes from right to left by adding `0`/`1` in the code table for the left/right subtree at the front of the codes.

a) Extend the module `Huffman` by a function `table2 : t -> table`, which generates the code table bottom to top.

b) Which version is more efficient?

4. (Exercise 4.6) Follow the computation of `sample ['h';'e';'l';'l';'o']` by evaluating (on paper) the results of all function calls starting at `sample ['h';'e';'l';'l';'o']`.

5. (Exercise 4.11) Implement depth-first-search and breadth-first-search for trees:

```
dfs : 'a tree -> 'a -> bool
bfs : 'a tree -> 'a -> bool
```

The functions should return true if and only if the tree contains the sought element.

⋆. (Exercise 4.12) Use search trees to implement the module `St` for finite sets where the signature is given by

```
type 'a t
val diff : 'a t -> 'a t -> 'a t
val empty : 'a t
val insert : 'a -> 'a t -> 'a t
val is_empty : 'a t -> bool
val mem : 'a -> 'a t -> bool
val of_list : 'a list -> 'a t
val singleton : 'a -> 'a t
val to_list : 'a t -> 'a list
val union : 'a t -> 'a t -> 'a t
```

i.e., internally the type `'a t` is `'a btree` as defined above but that fact is hidden from the user. The operations have following specifications (where S and T are sets and s and

[1] Here you really need the 6th edition of the course notes.

t are elements):

$$\begin{aligned}
\texttt{diff}\ S\ T &= S \setminus T \\
\texttt{empty} &= \varnothing \\
\texttt{insert}\ s\ S &= \{s\} \cup S \\
\texttt{mem}\ s\ S &= s \in S \\
\texttt{singleton}\ s &= \{s\} \\
\texttt{union}\ S\ T &= S \cup T
\end{aligned}$$