

The two problems below are (translations of) old examinations from Department of Computer Science, University of Copenhagen.

Problem 1

Consider the following declarations:

```
fun takewhile(_, []) = [] (* t1 *)
  | takewhile(p, x::xs) = if p x then x::takewhile(p,xs) else [] (* t2 *)

fun dropwhile(_, []) = [] (* d1 *)
  | dropwhile(p, x::xs) = if p x then dropwhile(p,xs) else x::xs (* d2 *)

infix @;
fun [] @ ys = ys (* @1 *)
  | (x::xs) @ ys = x::(xs@ys) (* @2 *)
```

Prove for all (terminating) predicates p and lists xs (of suitable types):

$$\text{takewhile}(p, xs) @ \text{dropwhile}(p, xs) = xs$$

Problem 2

Consider the declarations:

```
datatype 'a tr = L | B of 'a tr * 'a list * 'a tr;

fun tinliste L = []
  | tinliste(B(t1, xs, t2)) = tinliste t1 @ xs @ tinliste t2

fun tin xs L zs = xs @ zs
  | tin xs (B(t1, ys, t2)) zs = tin xs t1 (tin ys t2 zs);
```

Prove that

$$\text{tin } xs \ t \ zs = xs @ \text{tinliste } t @ zs$$

holds for all t, xs and zs (of suitable types).