University of Innsbruck
Test 1B

Institute of Computer Science
30 November 2007

Functional Programming WS 2007/2008 LVA 703018

Name: **MatrNr:** **StudienKZ:**

This test consists of four exercises. *Explain your answers.* The available points for each item are written in the margin.

[6] 1. Consider the function drop : int −> 'a list −> 'a list, defined by:

```
let rec drop n xs =
 if n <= 0 then xs else match xs with
  | [] −> []
  | _ :: xs −> drop (n − 1) xs
;;
```

Give at least 5 different intermediate steps of the derivation sequence starting at:

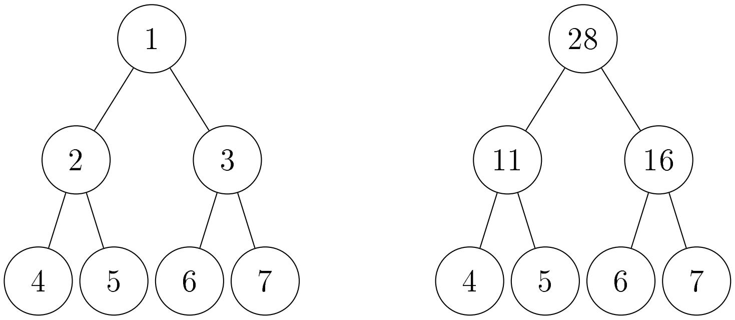
```
drop 2 [5; 7; 2; 4]
 →  if 2 <= 0 then [5; 7; 2; 4] else drop (2 − 1) [7; 2; 4]
```

[5] 2. Write a function times : int list −> int that computes the product of an integer list using the function fold : ('a −> 'b −> 'b) −> 'b −> 'a list −> 'b, defined by:

```
let rec fold f b = function
  | [] −> b
  | x :: xs −> f x (fold f b xs)
;;
```

[8] 3. Consider a binary tree (`type 'a tree = Empty | Node of ('a tree * 'a * 'a tree)`). Define a function `sum_tree : int tree -> int tree` that produces a tree where for each subtree t the root node of t is the sum of all nodes in t, i.e., changes the tree on the left into the one on the right.

Hint: An auxiliary function `sum_nodes : int tree -> int` that sums up the values of all nodes in a tree might be useful.



4. Consider the lambda-term $t = \lambda xw.(\lambda xz.y\ z)\ w\ z$.

[2] (a) Compute all subterms of t.

[2] (b) Compute $\mathcal{V}\mathsf{ar}(t)$, $\mathcal{BV}\mathsf{ar}(t)$, and $\mathcal{FV}\mathsf{ar}(t)$.

[2] (c) Is t in normal form?