RHEINISCH-
WESTFÄLISCHE
TECHNISCHE
HOCHSCHULE
AACHEN

LEHR- UND FORSCHUNGSGEBIET INFORMATIK II
RWTH Aachen · D-52056 Aachen · GERMANY
http://www-i2.informatik.rwth-aachen.de/lufgi2.html

Prof. Dr. Jürgen Giesl
Darius Dlugosz

Exercises *Functional Programming* – Sheet 2

Solutions will be collected until Thursday, May 10, 2001.

Exercise 1 (3 points)

Define the data structure $Integer$ for numbers of arbitrary precision as an instance of class Eq, Ord, Num and $Show$. The instance of the classes Eq shall contain the declaration for methods $(==)$, $(/=)$, Ord requires the methods $(<), (>), (<=), (>=), max$ and min, Num requires the methods $(+)$ (addition), $(-)$ (subtraction), $(*)$ (multiplication), $(/)$ (division), $negate$ (reverses the algebraic sign) and $Show$ the method $show$.

Hint: The proper representation of $Integer$ looks like $[Int]$.

Exercise 2 (1 point)

The data structures $Operator$ and $Term$ are defined as follows:

$$data\ Operator = Add \mid Mult$$

$$data\ Term = Const\ Float \mid Var\ String \mid BOp\ Operator\ Term\ Term$$

Define the data structure $Term$ as an instance of the classes Eq, Ord and $Show$ with methods analogous to exercise 1.

Exercise 3 (2 points)

For any list there exist the predefined functions $take, drop :: Int \to [a] \to [a]$, so that for any $n \in \mathbb{N}$ and any list xs we have:

$$take\ n\ [x_1, ..., x_m] = [x_1, ..., x_{min(n,m)}]$$

$$drop\ n\ [x_1, ..., x_m] = [x_{n+1}, ..., x_m]$$

$$take\ n\ xs\ \ +\!\!+\ drop\ n\ xs = xs$$

For the data structure

$$data\ Tree\ a = Node\ a\ [Tree\ a]$$

define the functions

$$prune :: Int \to Tree\ a \to Tree\ a$$

$$branches :: Int \to Tree\ a \to [[Tree\ a]]$$

$$joinT :: Tree\ a \to [[Tree\ a]] \to Tree\ a$$

that works in the same way on trees: $prune\ n$ cuts off all nodes that are deeper than n, $branches\ n$ collects the pruned branches of $prune\ n$ and for $joinT$, all $n \in \mathbb{N}$ and for any tree $tree$ we have:

$$joinT\ (prune\ n\ tree)\ (branches\ n\ tree) = tree$$

Exercise 4 (1 point)

Define the data structure $HetList$ with the constructors Nil and $Cons$, so that the following declarations doesn't return a type error.

$$hetlength\ Nil = 0$$

$$hetlength\ (Cons\ \_\ \ x) = 1\ +\ hetlength\ x$$

$$heterogeneous = Cons\ 1\ (Cons\ \ True\ \ (Cons\ \ 'X'\ \ Nil))$$

Give the type declaration of $hetlength$ and $heterogeneous$.

Exercise 5 (0.5 points)

Answer the question 5(c) from exercise-sheet 1.