Departement Informatica en Informatiekunde, Faculteit Bètawetenschappen, UU.
In elektronische vorm beschikbaar gemaakt door de TEX van A−Eskwadraat.
Het college INFOFP werd in 2004/2005 gegeven door Prof. dr. S.D. Swierstra.

Functioneel Programmeren (INFOFP)
21 maart 2005

The exam consists of three open questions (2 points each) and 4 multiple choice questions (1 point each). A wrong multiple choice answer will give a negative result ($-\frac{1}{4}$ point), whereas omitting the answer results in 0 points. Therefore, guessing is not recommended.

Opgave 1

Which of the following is True?

a) The data type *Maybe* represents the chance that the evaluation of an expression terminates.

b) *Maybe* takes two type arguments: one indicating the type of the value to return when the expression terminates and one if it does not.

c) Can be used to represent failure of some computation.

d) Cannot occur inside a list.

Opgave 2

What is the result of the following parser application: *many* (`symbol 'a'`) `äaa"`?

a) `äaa"`

b) (`äaa"`,)

c) [(`äaa"`,), (`äa"`, `ä"`), (`ä"`, `äa"`), (, `äaa"`)]

d) None of the above

Opgave 3

What is the correct definition of the function *segs* that returns all segments from a list?
For example: *segs* [2,3,4]= [[], [4], [3], [3,4],[2],[2,3],[2,3,4]].

a) $segs\ [] \quad = [[]]$
$segs\ (x : xs) = segs\ xs\ ++\ map\ (x :)\ (inits\ xs)$

b) $segs\ [] \quad = []$
$segs\ (x : xs) = segs\ xs\ ++\ map\ (x :)\ (inits\ xs)$

c) $segs\ [] \quad = [[]]$
$segs\ (x : xs) = map\ (x :)\ segs\ xs\ ++\ (inits\ xs)$

d) $segs\ xs = zipWith\ (++)\ (inits\ xs)\ (tails\ xs)$

Opgave 4

The function $intersperse :: a \to [a] \to [a]$ puts its first argument between all the elements of its second argument. Thus *intersperse* `'a' "xyz"` results in `"xayaza"`. Which is the correct definition?

a) $intersperse\ a\ as = tail\ .\ concat\ .\ map\ (\lambda x \to [a, x])\ \$\ as$

b) $intersperse\ a\ as = tail\ [(a : e)\ \mid\ e \leftarrow as\ ]$

c) $intersperse\ a\ as = foldr(\lambda e\ r \to (a : e : r))[]as$

d) $intersperse\ a\ as = foldr(\lambda r\ e \to (a : e : r))[]as$

Opgave 5

Write the function *split* that returns all possibilities of splitting a list in an element and the rest of the elements:

$$split\ [1, 2, 3, 4] = [(1, [2, 3, 4]), (2, [1, 3, 4]), (3, [1, 2, 4]), (4, [1, 2, 3])]$$

You may assume that all the elements in the argument list are different.

Opgave 6

Given the function $getInt :: IO\ Int$, which reads an integer value from standard input, **write a Helium program** that results in the following IO (the `3` has been typed in by the user):

```
Give a number: 3
1 * 3 = 3
2 * 3 = 6
3 * 3 = 9
...
10 * 3 = 30
Goodbye
```

Opgave 7

Given the data type:

```
data Tree = Fork a (Tree a) (Tree a)
          | Leaf
```

Write the function that returns a list of values stored in the nodes on one of the longest paths from the root to a leaf:

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