**EXAM FUNCTIONAL PROGRAMMING**
Tuesday the 1st of October 2013, 8.30 h. - 10.30 h.

| |
|---|
| Name: |
| Student number: |

**Before you begin**: Do not forget to write down your name and student number above. If necessary, explain your answers (in English or Dutch). For multiple choice questions, circle what you think is the best answer. Use the empty boxes under each question to write your answer and explanations in. Use the empty paper provided with this exam only as scratch paper (kladpapier). At the end of the exam, only hand in the filled-in exam paper. Answers will not only be judged for correctness, but also for clarity and conciseness. A total of one hundred points can be obtained; divide by 10 to obtain your grade. Good luck!

1. (i) What is the result of evaluating the following expression:

   (*foldr* (−) 0 . *map length*) `["Rood","Wit","Blauw"]`

   .../**8**

   (ii) The function *nrOfPositives* is defined as follows:

   *nrOfPositives* = *length* . *filter* (>0)

   Define this function with *foldr* by supplying the correct definitions of *e* and *op*, according to the following recipe:

   *nrOfPositives* = *foldr op e*
   **where** .....

   .../**15**

2. Consider the following datatype for representing boolean expression (propositions), where variable names consist of a single character:

$$\begin{aligned}\textbf{data}\ \mathit{Prop} = &\ \mathit{Cons\ Bool}\\ \mid &\ \mathit{Vari\ Char}\\ \mid &\ \mathit{Not\ Prop}\\ \mid &\ \mathit{Prop} :/\backslash: \mathit{Prop}\\ \mid &\ \mathit{Prop} :\backslash/: \mathit{Prop}\\ \mid &\ \mathit{Prop} :{-}{>}: \mathit{Prop}\end{aligned}$$

(i) Give the value of type *Prop* that represents the proposition $p \rightarrow (p \vee q)$

$\boxed{\ldots/\mathbf{6}}$

(ii) Write a function $\mathit{vars} :: \mathit{Prop} \rightarrow [\mathit{Char}]$ that takes a proposition and finds all the variables that occur in the proposition (you may choose to delete duplicates, or not).

$\boxed{\ldots/\mathbf{14}}$

(iii) On the next page, write an evaluator $\mathit{beval} :: \mathit{Prop} \rightarrow [(\mathit{Char}, \mathit{Bool})] \rightarrow \mathit{Maybe\ Bool}$. It takes a proposition and an environment *env*. The latter contains pairs of values, like (`'p'`, *True*), that give values to variables. In order to compute the truth value of the proposition, you must first verify that *env* has a truth value for every variable in that proposition (Hint: use (ii)). If this is not the case, you should return *Nothing*. Otherwise you should return *Just v* where *v* is the truth value computed for the proposition.

. . . /**14**

---

3. (i) Assuming that the prelude does not provide you with a definition for function composition ( . ), write your own. Also provide a type signature for ( . ) .

. . . /**9**

(ii) Explain what it means that you can regard ( . ) as a function that takes 1, 2 or 3 parameters.

. . . /**8**

(iii) Given the type signature $sum :: [Int] \to Int$, is the expression

$$\textbf{if } \mathit{True} \textbf{ then } [sum, length] \textbf{ else } [id]$$

type correct? If so, what is its type. If not, explain why not.

$\boxed{\ldots/\mathbf{10}}$

4. (i) Determine the type of *map map*. You should not just write down the type below, but also explain how you arrived at that type (for example, in the way that this is done in the lecture notes of this course).

$\boxed{\ldots/\mathbf{16}}$