

Parel 4
Functioneel Programmeren
Toets

25 september 2015

Alle opgaven tellen even zwaar.

U mag alleen voorgedefinieerde functies gebruiken die op dit toetsblad zijn afgedrukt (Z.O.Z.).

Opgave 1. Schrijf een functie `addPairs` die de opeenvolgende tweetallen van een lijst van getallen optelt. Het eerste en het laatste element moeten daarbij ook worden meegenomen, maar zonder er iets bij op te tellen (of door er nul bij op te tellen). Bijvoorbeeld:

```
addPairs [1,2,1] = [1,3,3,1]
addPairs [1,3,3,1] = [1,4,6,4,1]
```

Opgave 2. Gegeven is de volgende definitie van de functie `f`:

```
f p [] = ([], [])
f p (x:xs) | p x = (x:ys, zs)
            | otherwise = (ys, x:zs)
where
  (ys, zs) = f p xs
```

Stel dat `p` de eigenschap is die aangeeft of een getal even is of niet (dus bijvoorbeeld `p 4 = True`, en `p 5 = False`), wat is dan de uitkomst van de expressie

```
f p [1..10]
```

Opgave 3. Schrijf een functie `countDups` die telt hoeveel duplicaten van ieder element uit een lijst in die lijst voorkomt. Het resultaat moet als een lijst van 2-tupels worden opgeleverd van het element en het aantal keren dat het voorkomt. De volgorde waarin die 2-tupels in de resultaatlijst staan is niet van belang.

Bijvoorbeeld (voor een lijst van characters):

```
countDups "tartaar" = [('t',2), ('a',3), ('r',2)]
```

Opgave 4. Gegeven is een lijst van even lange lijsten getallen — dus als je de lijsten onder elkaar zou zetten, zou je een “rechthoek” van getallen krijgen. Schrijf een functie `addCols` die de overeenkomstige elementen uit de lijsten optelt, dus de functie `addCols` moet alle eerste elementen optellen, alle tweede elementen optellen, enzovoort. Dus als je de lijsten onder elkaar zou schrijven, moet de functie alle verticale kolommen van getallen optellen.

Bijvoorbeeld:

```
addCols [[1,2,3], [4,5,6], [7,8,9], [10,11,12]] = [22,26,30]
```

Lijst van toegestane operaties en functies

Getallen:	<code>+</code> , <code>-</code> , <code>*</code> , <code>/</code> , <code>^</code> , <code>==</code> , <code>/=</code> , <code>></code> , <code><</code> , <code>>=</code> , <code><=</code> <i>div</i> , <i>mod</i> , <i>abs</i> , <i>sqrt</i>
Characters:	<i>ord</i> , <i>chr</i>
Booleans:	<i>not</i> , <i>&&</i> , <i> </i>
Lijsten:	<code>:</code> , <code>++</code> , <code>\</code> , <i>head</i> , <i>tail</i> , <i>init</i> , <i>last</i> , <i>length</i> , <i>concat</i> , <i>reverse</i> , <i>elem</i> , <i>sum</i> , <i>minimum</i> , <i>maximum</i> , <i>take</i> , <i>drop</i> , <i>splitAt</i> , <i>zip</i> , <i>unzip</i>
Tupels:	<i>fst</i> , <i>snd</i>
Hogere orde:	<i>map</i> , <i>filter</i> , <i>zipWith</i>