

Parel 6
Functioneel Programmeren
Toets

11 oktober 2013

Alle opgaven tellen even zwaar.

Opgave 1. Schrijf een recursieve functie `pairs` die de elementen van een lijst groepeerd in 2-tupels van twee opeenvolgende elementen. Als het aantal elementen in de lijst oneven is, moet het laatste element twee keer opgenomen worden. Bijvoorbeeld:

```
pairs [1,2,3,4,5,6] = [(1,2),(3,4),(5,6)]
pairs [1,2,3,4,5,6,7] = [(1,2),(3,4),(5,6),(7,7)]
```

Opgave 2. Iemand wil een recursieve functie `remdups` (voor “remove duplicates”) schrijven die alle dubbele voorkomens van een element uit een lijst verwijdert. Bijvoorbeeld:

```
remdups [1,2,1,3,1,1,3,2] = [1,2,3]
```

Het basisgeval van de recursieve definitie is:

```
remdups [] = []
```

Het recursieve geval is

```
remdups (x:xs) = ...
```

Welke van onderstaande expressies moet hij kiezen voor het recursieve geval?

- a. `x : remdups [y | y <- xs , y /= x]`
- b. `remdups [y | y <- xs , y /= x]`
- c. `[y | y <- xs , y /= x]`
- d. `x : remdups xs`

Opgave 3. Schrijf een functie `doorsnede` die de doorsnede van twee lijsten bepaalt, dus die de lijst van elementen berekent die in beide gegeven lijsten zitten. Bijvoorbeeld:

```
doorsnede [1,3,5,7,9] [3,4,5,6,7] = [3,5,7]
```

Opgave 4. Gegeven is een lijst van n lijsten die allemaal n elementen bevatten — dus als je de lijsten onder elkaar zou zetten, zou je een “vierkant” van elementen krijgen. Schrijf een functie `diagonaal` die bij zo’n gegeven lijst van even lange lijsten de lijst van elementen oplevert die op de (hoofd)diagonaal (van linksboven naar rechtsonder) van dat vierkant staan. Bijvoorbeeld:

```
diagonaal [[1,2,3],[4,5,6],[7,8,9]] = [1,5,9]
```

