

Tentamen Functioneel Programmeren (211205)

2 juli 2009
9.00 – 12.30 uur

Opmerkingen vooraf:

- U mag het dictaat bij dit tentamen gebruiken, verder niets.
- Geef bij *elke* functie die u definieert het type.
- Beoordeling: er zijn drie opgaven, die allemaal even zwaar wegen.
- Succes!

Opgave 1.

- a. Schrijf een functie `isSorted` die test of een lijst getallen gesorteerd (van klein naar groot) is. De lijst kan meerdere copieën van hetzelfde getal bevatten.
- b. Schrijf een functie `tweeDelen` die test of een lijst van gehele getallen uit twee delen is opgebouwd: eerst alle oneven getallen, daarna alle even getallen. Zowel het even als het oneven deel (of beide) mag leeg zijn.
- c. Schrijf een functie `separate` die aan de hand van een gegeven eigenschap een lijst van elementen splitst in een paar van twee lijsten: eerst de lijst van elementen die de eigenschap hebben, daarna de lijst van elementen die de eigenschap *niet* hebben.
- d. Een matrix is een lijst van lijsten van getallen. Schrijf een functie `isVierkant` die test of een matrix vierkant is.
- e. Schrijf een functie `diag` die de lijst van getallen op de hoofddiagonaal (van links boven naar rechts onder) van een vierkante matrix oplevert.

Opgave 2. Gegeven is het volgende boomtype:

```
data Tree = Node [Int] [Tree]
```

Een boom van dit type heeft lijsten van getallen aan de nodes, en bij elke node een willekeurig aantal subbomen. U mag aannemen dat bij elke node zowel de lijst van getallen als de lijst van subbomen eindig zijn.

a. Schrijf een functie `vervang` die bij elke node de lijst van getallen vervangt door het middelste getal van die lijst. Gebruik hiervoor het `Maybe`-type en laat het middelste element van de lege lijst `Nothing` zijn. Definieer het boomtype dat u hierbij nodig hebt.

Bij een lijst van oneven lengte is het middelste getal eenduidig bepaald; neem bij een lijst van even lengte het getal net ná het midden van die lijst.

b. Definieer een geparametriseerd boomtype `ParTree` waarvan beide bij a gebruikte boomtypes instanties zijn.

Definieer beide bovengebruikte types als instanties van het type `ParTree`.

c. Schrijf een functie `langste` die de langste lijst getallen uit een boom oplevert (of één daarvan als er meerdere lijsten van dezelfde maximum lengte zijn).

d. Schrijf een functie `isSymm` die test of een boom symmetrisch is, d.w.z. van links af en van rechts af dezelfde boom is.

Opgave 3. Deze opgave gaat over gerichte grafen, waarvan de nodes aangegeven zijn door natuurlijke getallen (elke node uiteraard door een ander getal). Gegeven zijn de types

```
type Node = Int
type Graph = [ (Node, [Node]) ]
```

Het type `Graph` is een lijst van geordende paren (n, ms) , waarbij node n uitgaande edges heeft naar precies alle nodes in de lijst ms .

a. Schrijf een functie `bereikbaar` die, gegeven een gerichte graaf van type `Graph` en een `startnode`, de lijst van alle nodes oplevert die vanuit die `startnode` bereikbaar zijn. Daarbij mogen edges alleen in de goede richting worden doorlopen.

b. Een node n is een *wortel* van een graaf als alle nodes vanuit n bereikbaar zijn. Schrijf een functie `isWortel` die test of een node in een graaf een wortel is van die graaf.

c. De *ingraad* van een node is het aantal binnenkomende edges van die node. Schrijf een functie `ingraad` die de ingraad van een node bepaalt.

d. In een samenhangende graaf is een edge een *brug* als na verwijdering van die edge de graaf uit twee samenhangende subgrafen bestaat. Schrijf een functie `isBrug` die test of een edge in een graaf een brug is.

e. Vervang de eerste regel in bovenstaande typedefinitie door

```
data Node = Int
```

Is deze definitie correct? Is hij bruikbaar om een graaf te definiëren zoals hierboven bedoeld?