

Gequoteerde Haskell-oefening – (Studie van) Declaratieve Talen

NAAM:

RICHTING:

LOKAAL:

- Je krijgt twee uur om deze opdracht **individueel** op te lossen.
- Je raadpleegt enkel afgedrukte kopies van de slides en de cursusnota's (eventueel met handgeschreven nota's). Daarnaast mag de volgende elektronische documentatie raadplegen:
 - de beperkte library documentatie <http://zvon.org/other/haskell/Outputcomplex/index.html>
 - de ghci manual van jouw Haskell installatie; op de departementale machines staat die onder `///usr/share/doc/ghc-doc/html/libraries/base/index.html`
- In de map **1819_Gequoteerde/Haskell_Donderdag** op Toledo vind je de nodige bestanden evenals de indienmodule. Je vindt deze eveneens op **e-systant** onder **Haskell/Graded Sessions 18-19**.
- Je oplossing zet je in een bestand `Myhaskell.hs` en de eerste lijnen van dit bestand moeten je naam, studentnummer en richting bevatten.


```
-- Jan Jansen  
-- r0123456  
-- master cw
```
- Na twee uur, of wanneer je klaar bent, dien je het `Myhaskell.hs` bestand in via Toledo. **Let op: indienen via e-systant is niet voldoende.**

Pokémon GO Analyseren

Pokémon GO was de voorbije zomer een echte rage (voor sommigen nog steeds). Als je een smartphone hebt, is de kans groot dat je het zelf wel eens gespeeld hebt. Voor wie dit toch niet het geval is, volgt een korte samenvatting van de aspecten van het spel die relevant zijn voor deze opdracht: in Pokémon GO verschijnen (*spawn*) kleine wezentjes genaamd Pokémon op willekeurige momenten en willekeurige plaatsen in de echte wereld. Als speler loop je rond in de wereld met je smartphone in de hand, op zoek naar wilde Pokémon. Wanneer je een Pokémon tegenkomt, gooi je Pokéballs naar de Pokémon (door te swipen met je vinger) om de Pokémon te vangen. Een van de belangrijkste doelen in het spel is natuurlijk om van elke soort eentje te vangen (*catch 'em all*). Het spel bevat nog meer aspecten, bvb., vechten met Pokémon, . . . , maar deze zijn niet relevant voor deze opdracht.

Deze zomer, van maandag 5/9 tot zondag 11/9, heeft de Professor (Professor Oak, het personage van de bijbehorende Pokémon tv-reeks, niet professor Schrijvers of Janssens) de locaties en tijdstippen bijgehouden waarop Pokémon zijn verschenen op deze campus. De Professor heeft nu jou uitgekozen om deze data te analyseren om te zien of we interessante patronen kunnen vaststellen. Het is niet de bedoeling om statistisch relevante conclusies te trekken, we willen enkel maar enkele eenvoudige vragen beantwoorden, zoals: welke Pokémon verschijnen het vaakst? Waar verschijnen de meeste Pidgeys? Op welke weekdag verschijnen de meeste Pokémon? Hoeveel Pokémon verschijnen er overdag en hoeveel 's nachts?

Opdracht

- Implementeer de functies in de volgende opdrachten.
- Voor elke functie is er reeds een typesignatuur gegeven in de template.
Het is niet toegestaan de typesignaturen te wijzigen, maar je mag wel argumenten voor het gelijkheidsteken (=) zetten.
- De datatypes die gebruikt worden door de functies zijn gedefinieerd in de template, bekijk deze.
- Na opdracht 2 **kan je de volgorde kiezen** waarin je de overige opdrachten maakt.
- Het is toegelaten om hulpfuncties, -datatypes, -typeklassen, . . . te definiëren, meer zelfs, het wordt aangemoedigd.
- Het zal je hopelijk opvallen dat sommige opdracht op elkaar lijken. **Kopieer GEEN CODE, maar schrijf hogere-ordefuncties die je kan hergebruiken.**
- In de voorbeeldjes worden variabelen uit de template gebruikt voor de leesbaarheid. De uitvoer die je krijgt zal dus niet letterlijk overeenkomen met die uit de voorbeeldjes.

1 Projectiefuncties

```
spawnPokemon :: Spawn -> Pokemon
spawnLocation :: Spawn -> Location
spawnTime     :: Spawn -> Time
```

Implementeer de drie bovenstaande functies die een `Spawn` afbeelden op één van zijn drie parameters.

2 Groepeer een lijst van Spawns m.b.v. een functie.

```
groupSpawnsBy :: Eq k => (Spawn -> k) -> [Spawn] -> [(k, [Spawn])]
```

Dit is een hulpfunctie die het oplossen van de andere opdrachten makkelijker zal maken.

Voor elke `Spawn` in de lijst, roep de functie op met de `Spawn` om een `k` te krijgen. Maak een lijst met key-valueparen waarbij de keys al de verschillende `ks` zijn en de values lijsten van `Spawns` die dezelfde `k` hebben.

```
>>> groupSpawnsBy spawnPokemon testSpawns
[("Pidgey", [pidgeySpawn1, pidgeySpawn2]), ("Pikachu", [pikachuSpawn])]
```

3 Welke Pokémon verschijnen het vaakst?

```
mostCommonPokemon :: [Spawn] -> [(Pokemon, Int)]
```

Maak een lijst van `(pokemon, n)`-paren waarbij `n` het aantal keren is dat de `pokemon` verschenen is. Deze lijst is gesorteerd op `n`: het paar met de hoogste `n` komt eerst en het paar met de laagste `n` komt laatst.

Hint: kijk eens naar de `sortOn`-functie in de `Data.List`-module.

```
>>> mostCommonPokemon testSpawns
[("Pidgey", 2), ("Pikachu", 1)]
```

4 Op welke locatie verschijnt de gegeven Pokémon het vaakst?

```
topSpawnPointsOf :: Pokemon -> [Spawn] -> [(Location, Int)]
```

Maak een lijst van `(location, n)`-paren waarbij `n` het aantal keren is dat de gegeven Pokémon verschenen is op `location`. Deze lijst is gesorteerd op `n`: het paar met de hoogste `n` komt eerst en het paar met de laagste `n` komt laatst.

```
>>> topSpawnPointsOf "Pikachu" testSpawns
[((50.864605, 4.6786203), 1)]
```

5 Tijdens welke uren verschijnen de meeste Pokémon?

```
topHours :: [Spawn] -> [(Int, Int)]
```

Maak een lijst van `(hour, n)`-paren waarbij `n` het aantal keren is dat een Pokémon verschenen is tussen `[hour:00, hour:59]`. Deze lijst is gesorteerd op `n`: het paar met de hoogste `n` komt eerst en het paar met de laagste `n` komt laatst.

```
>>> topHours testSpawns
[(12, 2), (3, 1)]
```

6 Op welke weekday verschijnen de meeste Pokémon?

```
topWeekDays :: [Spawn] -> [(WeekDay, Int)]
```

Maak een lijst van (weekday, n)-paren waarbij n het aantal keren is waarop een Pokémon verschenen is op weekday. Deze lijst is gesorteerd op n: het paar met de hoogste n komt eerst en het paar met de laagste n komt laatst.

```
>>> topWeekDays testSpawns
[(Friday,2),(Tuesday,1)]
```

7 Hoeveel Pokémon verschijnen er overdag en hoeveel 's nachts??

```
dayAndNight :: [Spawn] -> (Int, Int)
```

Maak een paar met als eerste element hoeveel Pokémon er overdag verschenen zijn en als tweede element hoeveel Pokémon er 's nachts verschenen zijn. Met overdag bedoelen we [7:00, 21:00] (beide grenzen inbegrepen).

```
>>> dayAndNight testSpawns
(2,1)
```

8 Hoeveel Pokémon verschijnen er rond het uur en hoeveel tussen de uren?

```
aroundTheHours :: [Spawn] -> (Int, Int)
```

Maak een paar met als eerste element hoeveel Pokémon er rond het uur verschenen zijn en als tweede element hoeveel Pokémon er tussen de uren verschenen zijn. Met “rond het uur” bedoelen we van 15 minuten (inclusief 45) voor het uur tot 15 minuten na het uur (exclusief 15), of: [(x - 1):45, x:15[. Alle andere tijdstippen zijn “tussen de uren”, bvb., 12:30, 7:43, 18:16.

```
>>> aroundTheHours testSpawns
(2,1)
```

9 Analyseer de gegevens.

```
analyseSpawns :: IO ()
```

Laad de [Spawn] uit spawns.data. Het spawns.data-bestand bevat data die je kan omzetten van een String naar [Spawn] m.b.v. de read-methode van de Read-typeklasse.

Hint: kijk eens naar de readFile-functie in de System.IO-module.

Analyseer deze data m.b.v. de functies die je hebt geschreven en toon de resultaten via IO. De uitvoer moet er ongeveer zo uitzien:

```
>>> analyseSpawns
# Most common Pokemon:
1. Pidgey spawned 524 times
2. Rattata spawned 491 times
3. Weedle spawned 484 times

# Top spawn points of Pidgey:
1. at (50.863804,4.6802573) 40 Pidgeys spawned
2. at (50.864193,4.6786504) 38 Pidgeys spawned
```

3. at (50.86296,4.674903) 38 Pidgeys spawned

Top hours:

1. at 20 o'clock 180 Pokemon spawned
2. at 16 o'clock 180 Pokemon spawned
3. at 15 o'clock 180 Pokemon spawned

Top week days:

1. on Wednesday 720 Pokemon spawned
2. on Saturday 719 Pokemon spawned
3. on Friday 719 Pokemon spawned

More Pokemon spawn during the day than during the night: 2511 vs 1588

More Pokemon spawn around the hours than between the hours: 2596 vs 1503

N.B. de “top spawn points” worden getoond van de meest verschijnende Pokémon volgens de data. Pidgey is dus **niet** gehardcoded.

Wanneer er meer Pokémon 's nachts verschijnen dan overdag, verwissel de woorden “day” en “night” zodat je het volgende krijgt: “More Pokemon spawn during the night than during the day: x vs y”. Doe hetzelfde voor rond en tussen de uren.

Wanneer er evenveel Pokémon overdag als 's nachts verschijnen, toon dan “As many Pokemon spawn during the day as during the night: x”, waarbij “x” het aantal is. Gelijkaardig voor rond en tussen de uren: “As many Pokemon spawn around the hours as between the hours: x”.

In het geval van een ex-aequo, bvb., “Top hours”, is de volgorde niet van belang.