

Functioneel Programmeren

in vergelijking met andere paradigmas

Thomas van Binsbergen



De paradigma's

- ▶ Imperatief Programmeren (PHP, Javascript, Java, Python, Scala ...)
- ▶ Object Oriented programmeren (Java, PHP, Scala)
- ▶ Declaratief/Functioneel Programmeren (Haskell, Scala, ML)



Programmeren zonder assignment

- ▶ In de wiskunde zijn er veel variaties op het $=$ -teken
- ▶ PHP en Javascript zouden het $:=$ symbool moeten gebruiken
- ▶ De zogenoemde 'destructieve assignment'
- ▶ In Haskell bestaat deze operator niet
- ▶ Daardoor hebben we geen variabelen!



Wat hebben we wel?

- ▶ Functies met parameters
- ▶ Functies zonder parameters, constanten
- ▶ Datatypes (structuren)
- ▶ Recursie (om door structuren te reizen)
- ▶ Sterke (statische) Typering!



Zuiver programmeren (zonder side-effects)

Voordelen

- ▶ Side-effects vormen een grote bron voor fouten
- ▶ Eenvoudig redeneren over je programmas

Nadelen

- ▶ Stijlere leercurve
- ▶ In de praktijk hebben we side-effects:
- ▶ User-input, databases, file-input, willekeur, parallelisme



Statische Typering

Voordelen

- ▶ Veel fouten komen voor 'runtime' aan het licht
- ▶ Het type van een functie zegt veel over wat de functie doet
- ▶ Dient als een contract tussen gebruiker en implementeerder van de functie!

Nadelen

- ▶ De wereld van input/output is niet getypeerd, dit maakt parsen/printen noodzakelijk

Dynamische Typering

- In PHP gebeurt dit parsen/printen achter de schermen



Andere eigenschappen van Haskell

- ▶ Equational Reasoning
- ▶ Hogere-orde functies (abstractie! abstractie! abstractie!)
- ▶ Luie evaluatie
- ▶ Haskell is een uitstekende gastheer voor EDSLs



Waar wordt Haskell vooral voor gebruikt?

- ▶ Het schrijven van compilers (de ene taal naar de ander omzetten)
- ▶ Het analyseren en optimaliseren van broncode
- ▶ Grote financiële applicaties



Parsen, Transformeren, Printen

De pijpleiding

- ▶ Parsen: Input String \rightarrow Structuur
- ▶ Transformeren: Structuur \rightarrow Structuur
- ▶ Printen: Structuur \rightarrow Output String

