

Programmeren en Bewijzen in Dependently Typed Talen

Toon Nolten



Overzicht

- Dependent Types
- Agda en Haskell
- Gevalstudies
- Opmerkingen

Dependent Types

- Types kunnen afhangen van waarden

```
replicate : ∀ {A n} → A → Vec A n  
_!!_ : ∀ {A n} → Fin n → Vec A n → A
```

- Expressiviteit
- Meer verificatie door de typechecker

Agda en Haskell

- Een dependently typed taal tegenover een taal die in de praktijk wordt toegepast
- Agda is gericht op programmeren in tegenstelling tot andere dependently typed talen
- Haskell heeft geen dependent types maar GHC heeft wel extensies die het mogelijk maken gelijkaardige garanties te bekomen

Gevalstudies

- Koopa Troopas
 - Onmogelijk om verkeerde paden op te stellen
 - Een pad heeft een verschillend type naargelang de Koopa Troopa die er op loopt
- Red-Black Trees
 - Door een precieze definitie van de bomen kunnen we een belangrijke klasse fouten uitsluiten
 - Een deel van de mentale inspanning voor het schrijven van de code kunnen we aan de typechecker overlaten

Opmerkingen

- In Haskell kunnen we redelijk dicht bij dependent types komen, maar de code wordt wel ingewikkelder door de vermenigvuldiging van types
- Dependent types hebben niet alleen voordelen
 - Type inferentie wordt in het algemeen onmogelijk
 - Sommige argumenten zijn enkel nodig om de typechecker te overtuigen
- Talen met dependent types gericht op programmeren zijn nog jong en zullen naar de toekomst toe zeker nog verbeteren