

Seminarium: Programowanie w teorii typów

Przykładowe zastosowania typów zależnych

Wojciech Jedynek, Paweł Wieczorek

Instytut Informatyki Uniwersytetu Wrocławskiego

24 września 2011

- 1 Na początku był Haskell...
 - ...i wszystko było dobrze...
 - ...prawie wszystko.

Programowanie funkcyjne - mocne strony

- Programując w silnie typowanym języku funkcyjnym mamy gwarancję, że jeśli kompilator zaakceptuje nasz program, to wykluczone zostały pewne klasy błędów programistycznych
- Co z błędami logicznymi?
- Używając sprytu i doświadczenia możemy sobie pomóc...

Typ bool nie ma znaczenia!

W klasycznym programowaniu funkcyjnym typecheckerowi jest obojętne jaki będzie rezultat obliczenia predykatu w wyrażeniu warunkowym.

```
if null xs then
  head xs
else
  xs
```

“Well-typed programs can't go wrong”?

Informacyjność typów - podejście “tradycyjne”

W Haskellu piszemy:

```
sort :: Ord a => [a] -> [a]
```

Co możemy wywnioskować z takiej sygnatury? A czego **nie** możemy wywnioskować?

2 Typy zależne - motywacja

- Czy możemy lepiej?

Na czym polega programowanie z typami zależnymi?

- Programowanie w typami zależnymi opiera się na programowaniu funkcyjnym
- Dostajemy do dyspozycji znacznie bardziej rozbudowany system typów
- Główne hasło: *typ wyrażenia może zależeć od jego wartości*
- System wystarczająco silny, by mówić o **certyfikacji**

Wyrażanie własności danych

Systemy, którymi będziemy się tutaj zajmować pozwalają wyrażać własności takie jak:

- liczba n jest parzysta
- lista l jest posortowana
- drzewo t jest zbalansowane
- plik f jest otwarty

Informacyjność typów - typy zależne

Używając typów zależnych możemy napisać (pseudokod)

```
sort :: Ord a => (xs : [a]) ->  
    (ys : [a] && ordered ys && permutation_of xs ys)
```

Co możemy wywnioskować z takiej sygnatury?

Zastowanie praktyczne: projektowanie API

Gdy projektujemy interfejs dla biblioteki rzadko udaje się wyrazić wszystkie założenia w sygnaturach.

Przykład. Kółko i krzyżyk

Zaprojektuj interfejs dla operacji addMove i undo, zapytania empty? i wartości empty. Przeanalizuj relacje między nimi.

Projektowanie API c.d.

Stosując tradycyjne podejście pewne założenia musimy (w najlepszym wypadku) wyrazić jako komentarz...

Dobra wiadomość

Typy zależne pozwalają nam traktować dowody na równi z wartościami. Możemy programować i dowodzić w ramach tego samego systemu!

Uwagi

Żeby cała ta zabawa miała sens, logika, która jest wbudowana w język pozwalający na programowanie z typami zależnymi musi być niesprzeczna.

Większe przykłady

- Drzewa poszukiwań binarnych z niezmiennikiem
- Unifikacja pierwszego rzędu