

Metody programowania 2017

Lista zadań nr 8

Na zajęcia 25 i 26 kwietnia 2017

Zadanie 1 (1 pkt). Składnię abstrakcyjną drzew binarnych bez etykiet będziemy reprezentować w postaci struktur zbudowanych z atomu `leaf/0` i binarnego funktora `node/2`. Składnię konkretną takich drzew opisuje gramatyka

$$\begin{aligned}\langle \text{tree} \rangle &\rightarrow * \\ \langle \text{tree} \rangle &\rightarrow (\langle \text{tree} \rangle \langle \text{tree} \rangle)\end{aligned}$$

gdzie $*$ oznacza liść, a para drzew ujęta w nawiasy oznacza drzewo, w którym synami korzenia są podane drzewa. Napisz gramatykę DCG odpowiadającą powyższej gramatyce. Czy generuje ona wszystkie słowa należące do języka opisanego tą gramatyką? Przepisz gramatykę DCG tak, by otrzymany program prologowy działał deterministycznie podczas rozpoznawania słów (taki program nie nadaje się oczywiście do generowania słów). Dodaj odpowiednie akcje semantyczne budujące abstrakcyjne drzewa rozbioru.

Zadanie 2 (1 pkt). Oto gramatyka zawierająca binarny operator łączący w lewo:

$$\begin{aligned}\langle \text{expression} \rangle &\rightarrow \langle \text{simple expression} \rangle \\ \langle \text{expression} \rangle &\rightarrow \langle \text{expression} \rangle * \langle \text{simple expression} \rangle \\ \langle \text{simple expression} \rangle &\rightarrow a \\ \langle \text{simple expression} \rangle &\rightarrow b \\ \langle \text{simple expression} \rangle &\rightarrow (\langle \text{expression} \rangle)\end{aligned}$$

Napisz odpowiednią gramatykę DCG odpowiadającą powyższej gramatyce. Czy generuje ona wszystkie słowa należące do języka opisanego tą gramatyką? Przepisz gramatykę DCG tak, by otrzymany program prologowy działał deterministycznie podczas rozpoznawania słów (taki program nie nadaje się oczywiście do generowania słów). Dodaj odpowiednie akcje semantyczne budujące abstrakcyjne drzewa rozbioru takich wyrażeń, zbudowane z atomów `a/0` i `b/0` oraz binarnego funktora `*/2`.

Zadanie 3 (1 pkt). Wiedząc, że

$$\begin{aligned}(f . g) x &= f (g x) \\ \text{flip } f a b &= f b a \\ \text{curry } f a b &= f (a, b)\end{aligned}$$

zdefiniuj w Haskellu funkcję `swap` spełniającą następującą specyfikację:

$$\text{flip}(\text{curry } f) = \text{curry}(f . \text{swap})$$

dla każdego $f :: (a, b) \rightarrow c$.

Zadanie 4 (1 pkt). Wiedząc, że

$$\begin{aligned}(f . g) x &= f (g x) \\ f \$ x &= f x \\ \text{flip } f a b &= f b a \\ \text{head } (x:_) &= x \\ \text{tail } (_,xs) &= xs \\ \text{map } _ [] &= [] \\ \text{map } f (x:xs) &= f x : \text{map } f xs\end{aligned}$$

wyznacz typy następujących wyrażeń w Haskellu:

$$\begin{aligned}(&.)(&.) \\ (&.)(&\$) \\ (&\$)(&.) \\ \text{flip flip} \\ (&.)(&.)(&.) \\ (&.)(&\$)(&.) \\ (&\$)(&.)(&\$) \\ \text{flip flip flip} \\ \text{tail } \$ \text{map tail } [[],['a']]\end{aligned}$$
$$\begin{aligned}\text{let } x &= x \text{ in } x x \\ (\lambda _ \rightarrow 'a') (\text{head } []) \\ (\lambda (_,_) \rightarrow 'a') (\text{head } [])\end{aligned}$$

Do wyznaczenia typów nie używaj kompilatora!

Zadanie 5 (1 pkt). Wiedząc, że

$$\begin{aligned}(f . g) x &= f (g x) \\ \text{fst } (x,_) &= x \\ \text{snd } (_,y) &= y \\ \text{pair } (f,g) x &= (f x, g x) \\ \text{cross } (f,g) &= \text{pair } (f . \text{fst}, g . \text{snd})\end{aligned}$$

udowodnij, że:

$$\text{cross } (f, g) . \text{cross } (h, k) = \text{cross } (f . h, g . k).$$

Zadanie 6 (1 pkt). Mamy

```
class Enum a where
  fromEnum :: a -> Int
  toEnum :: Int -> a
```

Czy można zdefiniować typ (a, b) jako instancję klasy `Enum`, gdzie oba typy a i b są instancjami klasy `Enum`?

Zadanie 7 (1 pkt). Niech

```
data Nat = Zero | Succ Nat
(+), (*), (^) :: Nat -> Nat -> Nat
m + Zero = m
m + Succ n = Succ (m + n)
m * Zero = Zero
m * Succ n = (m * n) + m
m ^ Zero = Succ Zero
m ^ Succ n = (m ^ n) * m
```

Dla jakich wartości $x, m, n :: \text{Nat}$ jest spełniona następująca równość:

$$x ^ (m + n) = (x ^ m) * (x ^ n).$$