

Programowanie 2008

Egzamin, część II („magisterska”)

27 czerwca 2008, godzina 12:15

Punkty	Ocena
do 159	2.0
od 160 do 183	3.0
od 184 do 207	3.5
od 208 do 231	4.0
od 232 do 255	4.5
od 256 do 320	5.0

Należy zsumować punkty z pierwszej części egzaminu i punkty otrzymane za rozwiązanie poniższych zadań.

Czas trwania egzaminu: 150 minut

Zadanie 1 (80 pkt). Rozważmy lambda wyrażenia

$$t ::= x \mid \lambda x.t \mid t_1 t_2$$

z typami prostymi

$$\tau ::= o \mid \tau_1 \rightarrow \tau_2,$$

gdzie o jest stałą typową. Termom przypisujemy typy w zwykły sposób:

$$\frac{}{\Gamma, x : \tau \vdash x : \tau} \quad \frac{\Gamma, x : \tau_1 \vdash t : \tau_2}{\Gamma \vdash \lambda x.t : \tau_1 \rightarrow \tau_2} \quad \frac{\Gamma \vdash t_1 : \tau_2 \rightarrow \tau_1 \quad \Gamma \vdash t_2 : \tau_2}{\Gamma \vdash t_1 t_2 : \tau_1}$$

Liczebniki Churcha (zamknięte termy w postaci normalnej typu $o \rightarrow (o \rightarrow o) \rightarrow o$) reprezentują liczby naturalne zapisane w postaci unarnej. Podobnie zamknięte termy w postaci normalnej typu

$$o \rightarrow (o \rightarrow o) \rightarrow (o \rightarrow o) \rightarrow o$$

mogą reprezentować liczby naturalne w postaci binarnej (choć to bardzo niepraktyczna reprezentacja). Udowodnij, że zbiorem zamkniętych termów w postaci β -normalnej podanego wyżej typu (modulo α -konwersja) jest

$$\bigcup_{n=0}^{\infty} \{\lambda x w z. t \mid t \in B_n\},$$

gdzie

$$\begin{aligned} B_0 &= \{x\}, \\ B_{n+1} &= \{wt \mid t \in B_n\} \cup \{zt \mid t \in B_n\}. \end{aligned}$$

Zadanie 2 (80 pkt).

1. Przypomnij definicje klas: `Monad`, `MonadPlus` i `MonadTrans`.
2. Niech

```
class Monad m => MonadState s m | m -> s where
  get :: m s
  put :: s -> m ()
```

Zdefiniuj typ `State s` a reprezentujący obliczenie ze stanem `s` i wynikiem `a` i uczyn typ `State s` instancją klas `Monad` oraz `MonadState`.

3. Uczyn typ `[]` instancją klas `Monad` i `MonadPlus`.
4. Zdefiniuj transformator monad `StateT s m` a dodający do dowolnej monady `m` funkcjonalność monady stanowej ze stanem `s` i wynikiem `a`. Uczyn typ `StateT s m` instancją klas `Monad` (gdy `m` jest monadą), `MonadPlus` (gdy `m` jest monadą z plusem) i `MonadState` (gdy `m` jest monadą) a typ `StateT s` — instancją klasy `MonadTrans`.
5. Zdefiniuj transformator monad `ListT m` a dodający do dowolnej monady `m` funkcjonalność listy. Uczyn typ `ListT m` instancją klas `Monad` i `MonadPlus` (gdy `m` jest monadą) i `MonadState` (gdy `m` jest monadą stanową), a typ `ListT` — instancją klasy `MonadTrans`.
6. Niech

```
data Id a = Id a
```

Uczyn typ `Id` monadą.

7. Niech

```
type Parser s = ListT (StateT s Id)
```

Zdefiniuj parser

```
hasProp :: (t -> Bool) -> Parser t t
```

sprawdzający, czy pierwszy token spełnia podaną własność i zwracający wówczas ten token oraz kombinatory parserowe odpowiadające konkatenacji ($<.>$), alternatywie ($<|>$) i domknięciu Kleene'go (**many**).

8. Niech

```
type Parser s = StateT s (ListT Id)
```

Wykonaj dla tego typu polecenie z poprzedniego punktu. W którym miejscu pojawiają się problemy, a w którym ta reprezentacja jest lepsza?

Zadanie 3 (80 pkt). Niech X będzie pewnym zbiorem. Udowodnij, że relacja inkluzji jest porządkiem zupełnym na zbiorze $X^{\subseteq X}$ funkcji częściowych określonych na zbiorze X .

W pierwszej części egzaminu rozważaliśmy instrukcję **repeat** c **until** b . Jej denotację możemy zdefiniować jako najmniejszą funkcję $\llbracket \text{repeat } c \text{ until } b \rrbracket$ spełniającą odpowiednią zależność rekurencyjną. Przypomnij tę zależność.

Niech Π będzie przestrzenią stanów maszyny. Zdefiniuj funkcjonal $\Phi : \Pi^{\subseteq \Pi} \rightarrow \Pi^{\subseteq \Pi}$ o tej własności, że $h \in \Pi^{\subseteq \Pi}$ spełnia podaną przez Ciebie zależność rekurencyjną wtedy i tylko wtedy, gdy jest punktem stałym Φ . Pokaż, że Φ jest ciągły. Wywnioskuj stąd (przytocz odpowiednie twierdzenie), że definicja denotacji **repeat** jest poprawna.