Teoretyczne podstawy informatyki
zadanie 50

Jarosław Socha

17 czerwca 2024

1 Treść zadania

Zadanie 50
Zakładając, że mamy λ-wyrażenia zdefiniowane na wykładzie oraz `plus` (dodawanie), `mult` (mnożenie) i `iszero` zdefiniuj wyrażenia `pred` (poprzednik), `minus` (odejmowanie), `leq` (mniejsze lub równe), `le` (mniejsze) i `iseq` (równe) na liczebnikach Church'a. Poprzednikiem zera jest zero, wynikiem odejmowania liczby większej od mniejszej jest zero.

2 Rozwiązanie

2.1 poprzednik - *pred*

Aby zdefiniować *pred* zdefiniujemy najpierw kilka wyrażeń pomocniczych. Po pierwsze, stworzymy wyrażenie *pair*, które będzie odpowiadało parze wyrażeń (w naszym przypadku liczb) i wyrażenia $first$ i *second* które będzie wyciągało odpowiednio element pierwszy i drugi wyrażenia.

$$pair = \lambda ab.(\lambda x.ab)$$

$$first = \lambda p.p(\lambda ab.a)$$

$$second = \lambda p.p(\lambda ab.b)$$

Następnie stworzymy funkcję *step*, która dla pary liczb (a, b) zwróci parę liczb $(b, b + 1)$

$$step = \lambda p.pair\,(second\,p)\,(succ\,second\,p)$$

gdzie *succ* to wyrażenie następnika $\lambda nfx.nf(fx)$.

Funkcja poprzednika będzie odpowiadała operacji wzięcia pierwszego elementu z pary $(0, 0)$, na którą n razy nałożymy funkcję *step*.

$$pred = \lambda n.first(n\,step\,(pair\,\underline{0}\,\underline{0}))$$

gdzie $\underline{0}$ to 0 w formie liczebników Church'a. Poprzednikiem zera jest zero.

2.2 odejmowanie - *minus*

Odejmowanie zdefiniujemy podobnie do dodawania za pomocą funkcji następnika, ale użyjemy poprzednika. *plus* wygląda tak:

$$plus = \lambda mn.n\, succ\, m$$

Więc *minus* będzie wyglądał następująco:

$$minus = \lambda mn.n\, pred\, m$$

Gdzie odejmując liczbę większą od mniejszej otrzymamy 0 przez konstrukcję poprzednika.

2.3 mniejszości - *le* i *leq*

Warunek mniejszości $n \leqslant m$ jest równoważny $n - m \leqslant 0$, a dla liczb naturalnych $n - m = 0$, czyli wystarczy sprawdzić, czy wynik odejmowania jest równy zero. Wyrażenie będzie zatem wyglądało:

$$leq = \lambda nm.iszero(minus\, n\, m)$$

Warunek mniejszości $n < m$ na liczbach naturalnych odpowiada warunkowi $n + 1 \leqslant m$, czyli wyrażenie wygląda następująco:

$$le = \lambda nm.leq(succ\, n)m$$

2.4 równość - *iseq*

Równość $n = m$ możemy wyrazić za pomocą poprzednich wyrażeń jako $n \leqslant m\ and\ m \leqslant n$. Otrzymamy wtedy:

$$iseq = \lambda nm.and(leq\, n\, m)(leq\, m\, n)$$