λ–rachunek c.d.

Inspiracja

Materiały do ćwiczeń będą oparte w dużej mierze na materiałach dr. Sławomira Bakalarskiego (do tego samego kursu).

Liczby naturalne

Liczby naturalne możemy reprezentować w rachunku λ w następujący sposób (są to tak zwane *numerały Churcha*):

$\underline{0} = \lambda sx.x$

$\underline{1} = \lambda sx.sx$

$\underline{2} = \lambda sx.s(sx)$

$\underline{3} = \lambda sx.s(s(sx))$

$\vdots$

W szczególności $\underline{n} = \lambda sx.s(\ldots(sx)\ldots)$, gdzie po prawej stronie $\lambda sx.$ zmienna s występuje dokładnie n razy.

Zauważmy też, że $\underline{n}fx$ odpowiada wywołaniu n razy f na x.

Reprezentacje funkcji (za wykładem)

Term E *reprezentuje* funkcję $f : \mathbb{N}^k \to \mathbb{N}$ jeśli term

$$E \; \underline{n_1} \; \dots \underline{n_k}$$

da się zredukować do

$$\underline{f(n_1, \dots, n_k)}.$$

Przykład: Funkcja k zmiennych stale równa n jest reprezentowana przez term $T = \lambda n_1 \dots n_k sx.s(\dots(sx)\dots)$, gdzie liczba s po prawej stronie $\lambda n_1 \dots n_k sx.$ wynosi dokładnie n. Zauważmy, że używając k razy β redukcji do T zaaplikowanego do k termów, te k termów zostanie "zignorowane" i dostaniemy $\underline{n}$.
Funkcję, dla której istnieje term, który ją reprezentuje nazywamy *definiowalną w λ–rachunku*.

Rekurencja: Operator punktu stałego (za wykładem)

Ważnym narzędziem w reprezentowaniu funkcji danych rekurencyjnie jest *operator punktu stałego*. Powiemy, że term $\mathcal{Y}$ jest operatorem punktu stałego jeśli dla dowolnego termu M zachodzi

$$\mathcal{Y}M \to_\beta M(\mathcal{Y}M).$$

Przykładem takiego termu jest

$$\mathcal{Y} = \lambda y.(\lambda x.y(xx))(\lambda x.y(xx)).$$

Rekurencja: Silnia

Używając operatora punktu stałego możemy zdefiniować silnię w następujący sposób:

$f = \lambda gn.if\_then\_else(ZERO\ n)(\underline{1})(MUL\ n\ (g(PRED\ n)))$

$FACT = \mathcal{Y}f$

Wyrażenia lambda w Haskellu

W Haskellu term $\lambda xy.F$ reprezentujemy przez `\x y -> F`.
Na przykład `\x y -> x*x + y*y` jest funkcją, która przyjmuje dwa argumenty i zwraca sumę ich kwadratów.

Bibliografia (do materiałów dr. S. Bakalarskiego, na których się opierałem)

- Paul Hudak.
  A Brief and Informal Introduction to the Lambda Calculus.
  url:
  http://www.cs.yale.edu/homes/hudak/CS201S08/lambda.pdf.
- Bruce J. MacLennan.
  Functional programming practice and theory. Addison-Wesley, 1990.
- Wazniak. Paradygmaty programowania. url:
  http://wazniak.mimuw.edu.pl/index.php?title=Paradygmaty_programowania.
- Wikipedia. Fixed-point combinator. url: https://en.wikipedia.org/wiki/Fixed-point_combinator#Fixed_point_combinators_in_lambda_calculus.
- Wikipedia. Rachunek lambda. url:
  https://pl.wikipedia.org/wiki/Rachunek_lambda.