Základy λ-kalkulu

doc. Dr. Ing. Miroslav Beneš
katedra informatiky, A-1007
☎ 59 732 4213

λ-kalkul

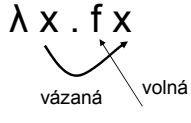
- 1930 Alonzo Church
  - netypovaný λ-kalkul
  - matematická teorie funkcí
- Základ všech funkcionálních jazyků
- Některé konstrukce i v imperativních jazycích (např. Python)

Syntaxe λ-kalkulu

- Proměnné
  - x, y, z, f, g, …
- λ-abstrakce
  - (λx . e)
- Aplikace
  - (e_1 e_2)
- Konvence pro závorky
  - λx . λy . e_1 e_2 = (λx . (λy . e_1 e_2))
  - e_1 e_2 e_3 = ((e_1 e_2) e_3)

Proměnná

- označuje libovolnou hodnotu
- v daném kontextu označuje vždy tutéž hodnotu (neexistuje možnost přiřazení)
- vázaná a volná proměnná



λ-abstrakce

- λ *x* . *e*
  - funkce s parametrem *x* a tělem *e*
- λ x y . e
  - funkce s parametry *x*, *y* a tělem *e*
  - ekvivalentní zápisu λ *x* . (λ *y* . *e*)
- λ e . e (λ f x (f x x)) (λ f x (f x x))

Aplikace

- (e_1 e_2)
  - aplikace funkce e_1 na argument e_2
- (*f x y*)
  - aplikace funkce (*f x*) na argument *y*
  - aplikace funkce *f* na argumenty *x* a *y*

Substituce

- e_1 [e_2/x]
  - nahrazení všech volných výskytů proměnné *x* ve výrazu e_1 za výraz e_2
  - substituce musí být *platná* (volná proměnná ve výrazu e_2 se nesmí stát vázanou)
- (λ x y . f x y) [g z / f] = λ x y . (g z) x y
- (λ x y . f x y) [g z / x] = λ x y . f x y
- (λ x y . f x y) [g y / f] = *není platná subst.*

Vyhodnocování λ-výrazů

- α-redukce
  - λ x . e ↔ λ y . e[y / x]
  - přejmenování vázané proměnné
  - substituce musí být platná
- β-redukce
  - (λ x . e_1) e_2 ↔ e_1 [e_2 / x]
  - "volání funkce" – nahrazení parametru hodnotou argumentu
  - substituce musí být platná

Vyhodnocování λ-výrazů

- η-redukce
  - λ x . f x ↔ f
  - odstranění abstrakce
  - proměnná *x* nesmí být volná v *f*
  - *Funkce jsou ekvivalentní, pokud pro všechny hodnoty parametrů dávají tentýž výsledek.*
  - umožňuje definovat *řezy* funkcí

    (+1) = λ x . (x + 1)

Příklady

- (λ f x . f x x) (λ x y . p y x)
  $=_\beta$ λ x . (λ x y . p y x) x x
  $=_\alpha$ λ z . (λ x y . p y x) z z
  $=_\beta$ λ z . (λ y . p y z) z
  $=_\beta$ λ z . p z z

- (λ f x . f x x) (λ x y . p y x)
  $=_\eta$ (λ f x . f x x) (λ y . p y)
  $=_\eta$ (λ f x . f x x) p
  $=_\beta$ λ x. p x x

Normalizační teorémy

- *redex* --- **red**ucible **ex**pression
  - výraz, který lze dále redukovat; α-redex, β-redex
- *normální forma výrazu*
  - výraz neobsahuje žádný β-redex

- Church-Rosserovy teorémy
  - Pokud e1 ↔ e2, pak existuje výraz e takový, že
    e1 → e **a** e2 → e
  - Pokud e1 → e2 a e2 je v normální formě, pak existuje redukční posloupnost z e1 do e2 (*normální redukční posloupnost*)

- Pokud existuje normální forma, lze k ní dojít normální redukční posloupností (leftmost outermost redex)