

Antassi del λ -calcolo

Luca Padovani

Linguaggi e Paradigmi di Programmazione

È proibito condividere e divulgare in qualsiasi forma i materiali didattici caricati sulla piattaforma e le lezioni svolte in videoconferenza. Ogni azione che viola questa norma sarà denunciata agli organi di Ateneo e perseguita a termini di legge.

perché studiare il λ -calcolo in questo corso?

- 1 definire in modo preciso il processo di valutazione (**riduzione**) di un programma funzionale
- 2 fare pratica di programmazione con funzioni a un solo argomento (**currificate**) e con **funzioni di ordine superiore**
- 3 definire in modo preciso cosa distingue i linguaggi **eager** (in pratica tutti quelli esistenti tranne Haskell) da quelli **lazy** (Haskell)
- 4 preparare le basi per il successivo sviluppo di un semplice **sistema di tipi** ed il relativo **algoritmo di inferenza**

due punti di vista sulle funzioni

Punto di vista estensionale

$$f = \{(0, 1), (1, 2), (2, 5), (3, 10), \dots\}$$

- + elegante interpretazione insiemistica
- non dice **come** calcolare f (e non è detto che sia possibile)
- non dice **quanto costa** calcolare f (ammesso che sia possibile)

Punto di vista intensionale

$$f(x) = x^2 + 1$$

- + abbiamo una “ricetta” che spiega come calcolare $f(x)$ noto x
- non tutte le funzioni hanno una ricetta (finita)

Il λ -calcolo è un linguaggio per scrivere “ricette” di funzioni

sintassi delle λ -espressioni (o λ -termini)

Variabili

- Sia $Var = \{x, y, z, \dots\}$ un insieme *infinito* di **variabili**

Sintassi

$$M, N ::= x \mid (\lambda x.M) \mid (M N)$$

Terminologia

- $(\lambda x.M)$ è un'astrazione o funzione con **argomento** x e **corpo** M
- $(M N)$ è l'applicazione della **funzione** M all'**argomento** N

Esempi

$$(\lambda x.x) \quad ((\lambda x.(x x)) (\lambda y.(y y))) \quad (\lambda f.(\lambda x.(f (f x))))$$

convenzioni sintattiche

Per limitare l'uso di parentesi e migliorare la leggibilità:

- ▶ le parentesi più esterne sono omesse

$$M\ N = (M\ N)$$

- ▶ il corpo delle astrazioni si estende a destra il più possibile

$$\lambda x.x\ x = (\lambda x.(x\ x)) \qquad \lambda x.x\ x \neq ((\lambda x.x)\ x)$$

- ▶ l'applicazione è associativa a sinistra

$$M_1\ M_2\ M_3 = ((M_1\ M_2)\ M_3)$$

esercizi

Mi devo abituare subito a giustificare i passi in modo formale

Rimuovere il più possibile le parentesi, senza cambiare il significato

1 $(\lambda x.((x x) x))$ $\Delta v ((v v) v)$

2 $(\lambda x.(\lambda y.(\lambda y.((x z) (y z)))))$ $\Lambda v (\Lambda v (\Lambda v ((v$

3 $(((a b) (c d)) ((e f) (g h)))$ $((a b) (c d)) ((e$

Ripristinare tutte le parentesi omesse, senza cambiare il significato

1 $x x x x$ $(((v v) v) v)$

2 $\lambda x.x \lambda y.y$ $(\Delta v (v (\Delta v v)))$

3 $\lambda x.(x \lambda y.y x x) x$ $(\Lambda v (v (\Lambda v v) v)$