Linguaggi di programmazione modulo 2
Teoria

Carlo Ramponi

May 9, 2019

Lamda Calcolo

Introduzione

In ML: $fn\ x => e$, in lamda calcolo: $\lambda x.e$
Serve per formalizzare quello detto fin'ora sulla programmazione funzionale.
È stata inventata molto prima per studiare la computabilità!

Definizione formale

Un' **espressione** nel lamda calcolo è un nome, una funzione o un' applicazione di espressioni a espressioni

- **Funzione:** $\lambda nome.espressione$
- **Applicazione:** $espressione_1\ espressione_2$
  Dove *espressione* è:
  - $x \rightarrow$ Un identificatore (variabile o costante)
  - $\lambda x.e \rightarrow$ Un'altra funzione
  - $e\ e \rightarrow$ Una generica espressione

Variabili libere e variabili legate

Def. Una variabile libera all'interno di un'espressione e è una variabile alla quale non è ancora stato assegnato un valore.
$F_v(e) \rightarrow$ "variabili libere nell'espressione e"
Def. Una variabile legata all'interno di un'espressione e è una variabile alla quale è stato assegnato un valore (una funzione o un'altra espressione)
$B_v(e) \rightarrow$ "variabili legate nell'espressione e"

Conseguenze

- $F_v(\lambda x.e) = F_v(e) \setminus \{x\}$
- $B_v(\lambda x.e) = B_v(e) \cup \{x\}$

L'operatore $\lambda x.e$ rimuove una variabile libera dall'espressione e e la aggiunge a quelle legate

Sostituzione

Nel lamda calcolo è possibile sostituire un identificatore con un altra espressione,

$$e[e'/x]$$

significa sostituire nell'espressione e tutte le occorrenze della x con e', che potrebbe essere un'altra espressione.
Il risultato della sostituzione si indica con " $\rightarrow$ "

Conseguenze banali:

- Se x è un identificatore, allora $x[e'/x] = e'$
- Se $x \neq y$, allora $y[e'/x] = y$

Applicazione:

$$(e_1\ e_2)[e'/x] = (e_1[e'/x]\ e_2[e'/x])$$

Astrazione:

- Se $x \neq y$ e $y \notin F_v(e')$, allora $(\lambda y.e)[e'/x] = (\lambda y.e[e'/x])$
- Se $x = y$, allora $(\lambda y.e)[z/x] = (\lambda y.e)$

Equivalenza di espressioni

- Date due espressioni e_1 ed e_2, quando si possono dichiarare equivalenti? Quando differiscono solo per il nome di variabili legate!
- Se y non è presente in e,

$$\lambda x.e \equiv \lambda y.e[y/x]$$

- Questa relaz. di equivalenza si chiama α-equivalenza e si indica con $\equiv_\alpha$
- Due espressioni si dicono α-quivalenti se una si può ottenere dall'altra sostituendo parte della prima con una α-equivalente

β-equivalenza

Intuitivamente, significa valutare le funzioni in un'espressione, formalmente

$$(\lambda x.e)e' \rightarrow_\beta e[e'/x]$$

Terminologia:

- $(\lambda x.e)e'$ è un "redex" (espressione riducibile)
- Si riduce a $e[e'/x]$

Osservazione: le β-riduzioni non sono simmetriche

$$(e_1 \rightarrow_\beta e_2) \not\Rightarrow (e_2 \rightarrow_\beta e_1)$$

Quindi non è una relazione di equivalenza.
Informalmente, $e_1 =_\beta e_2$ significca che esiste una sequenza di β-riduzioni da e_1 a e_2

Forme Normali

Un'espressione che n on contiene redex non ha β-riduzioni.

- Questa è chiamata "forma normale"
- $\lambda x.\lambda y.x$ è una forma normale
- $\lambda x.(\lambda y.y)x$ non è in forma normale, perchè:
  $(\lambda y.y)x \rightarrow_\beta x$ quindi $\lambda x.(\lambda y.y)x \rightarrow_\beta \lambda x.x$
- Le β-riduzioni possono terminare in una forma normale.
- O potrebbero andare avanti per sempre (come la ricorsione infinita o cicli infiniti):

$$(\lambda x.xx)(\lambda x.xx) \rightarrow_\beta (xx)[\lambda x.xx/x] = (\lambda x.xx)(\lambda x.xx)$$

Confluenza(?)

Teorema Se e può essere ridotto a e_1 con una β-riduzione e e può essere ridotto a e_2 con una β-riduzione, allora esiste un'espressione e_3 tale che sia e_1 che e_2 possono essere ridotte a e_3 con una β-riduzione.
Ciò significa che se e può essere ridotto ad una forma normale allora l'ordine delle riduzioni non importa.