calcolo con tipi

Luca Padovani
Linguaggi e Paradigmi di Programmazione

errori di programmazione

- sebbene il λ calcolo sia computazionalmente completo, per ragioni di **efficienza**, ogni linguaggio di programmazione basato sul λ calcolo deve fornire dati nativi (numeri, valori booleani, caratteri, ecc.) e le corrispondenti operazioni (somma, congiunzione, ecc.)
- non appena si fa ciò, si pone immediatamente il problema di gestire espressioni **sintatticamente corrette** ma **prive di significato** (la somma di un numero e di un booleano, la congiunzione logica di due numeri, ecc.)
- inoltre, questi **errori di programmazione** sono presenti anche nel λ-calcolo pure, dove però sono "nascosti" dal fatto che lì si possono solo definire e applicare funzioni
- usiamo i **tipi** per individuare (alcuni) errori

sintassi delle λ-espressioni con booleani

Estendiamo la sintassi con le costanti booleane e l'$\mathtt{if}$

$$
\begin{array}{llcll}
\textbf{Espressioni} & M, N & ::= & x & \text{variabile} \\
 & & \mid & \mathtt{c} & \text{costante} \\
 & & \mid & \lambda x.M & \text{astrazione} \\
 & & \mid & M\ N & \text{applicazione} \\
 & & \mid & \mathtt{if}\ M\ N_1\ N_2 & \text{condizionale} \\
 & & & & \\
\textbf{Costanti} & \mathtt{c} & \in & \{\mathtt{False}, \mathtt{True}\} &
\end{array}
$$

Estendiamo la semantica con riduzioni per l'$\mathtt{if}$

$$
\begin{array}{rcl}
\mathtt{if}\ \mathtt{True}\ M\ N & \to & M \\
\mathtt{if}\ \mathtt{False}\ M\ N & \to & N
\end{array}
$$

Molte espressioni sono sintatticamente corrette ma prive di senso

- $\mathtt{if}\ (\lambda x.x)\ M\ N \not\to$
- $\mathtt{True}\ \mathtt{False} \not\to$
- ...

tipi

- possiamo individuare questi errori dando un **tipo** alle espressioni
- un tipo è una forma sintattica di classificazione delle espressioni

Obiettivi

- la forma normale di una espressione di tipo `Bool`, se esiste, è una costante booleana (`False` o `True`)
- la forma normale di una espressione di tipo $t \to s$, se esiste, è un'astrazione che, applicata ad una espressione di tipo t, produce una espressione di tipo s

Sintassi dei tipi

Tipi	t, s	$::=$	`Bool`	booleani
		$\mid$	$t \to s$	funzioni

giudizi

Giudizio: M è ben tipato e ha tipo t $\qquad \vdash M : t$

- in generale M conterrà variabili libere
- occorre relativizzare il tipo di M al tipo delle variabili libere di M
- introduciamo contesti per tracciare il tipo delle variabili libere di M

Giudizio: M è ben tipato e ha tipo t nel contesto Γ $\qquad \Gamma \vdash M : t$

Definizione (contesto)

Un ***contesto*** *Γ è una funzione parziale da variabili a tipi.*

Notazione

- Scriviamo $dom(\Gamma)$ per il dominio di Γ
- Scriviamo $x : t$ per il contesto Γ tale che $dom(\Gamma) = \{x\}$ e $\Gamma(x) = t$
- Scriviamo Γ, Γ' per l'unione di Γ e Γ' quando $dom(\Gamma) \cap dom(\Gamma') = \emptyset$

regole di tipo

Forma generale di una regola

$$\frac{premessa_1 \quad \cdots \quad premessa_n}{conclusione}\ [\text{nome regola}]$$

- "se le premesse sono vere, allora la conclusione è vera"
- una regola senza premesse è detta **assioma**

Regole di tipo per il λ-calcolo con costanti

$$\frac{}{\Gamma, x : t \vdash x : t}\ [\text{t-var}] \qquad \frac{}{\Gamma \vdash \mathtt{c} : \mathtt{Bool}}\ [\text{t-bool}] \qquad \frac{\Gamma, x : t \vdash M : s}{\Gamma \vdash \lambda x.M : t \to s}\ [\text{t-lam}]$$

$$\frac{\Gamma \vdash M : \mathtt{Bool} \qquad \Gamma \vdash N_1 : t \qquad \Gamma \vdash N_2 : t}{\Gamma \vdash \mathtt{if}\ M\ N_1\ N_2 : t}\ [\text{t-if}] \qquad \frac{\Gamma \vdash M : t \to s \qquad \Gamma \vdash N : t}{\Gamma \vdash M\ N : s}\ [\text{t-app}]$$

proprietà delle espressioni ben tipate

Lemma (subject reduction)

Se $\Gamma \vdash M : t$ e $M \rightarrow N$ allora $\Gamma \vdash N : t$.

Definizione (valore)

*Diciamo che M è un **valore** se M è una costante o un'astrazione.*

Esempi

- $(\lambda x.x)$ `True` non è un valore (si riduce)
- `True` $(\lambda x.x)$ non è un valore (non si riduce)
- `if` $(\lambda x.x)$ `True` `False` non è un valore (non si riduce)
- $\lambda x.(\lambda y.y)$ `True` è un valore

Teorema (progresso)

Se $\vdash M : t$ e $M \Rightarrow N \not\rightarrow$ allora N è un valore.

esempio $(\lambda x.x)\ \mathtt{False}$

$$\dfrac{\dfrac{\dfrac{}{x : \mathtt{Bool} \vdash x : \mathtt{Bool}}\,[\text{t-var}]}{\vdash \lambda x.x : \mathtt{Bool} \to \mathtt{Bool}}\,[\text{t-lam}] \qquad \dfrac{}{\vdash \mathtt{False} : \mathtt{Bool}}\,[\text{t-bool}]}{\vdash (\lambda x.x)\ \mathtt{False} : \mathtt{Bool}}\,[\text{t-app}]$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x : t \vdash x : t}\,[\text{t-var}]}{\vdash \lambda x.x : s}\,[\text{t-lam}] \qquad \dfrac{\dfrac{}{y : \texttt{Bool} \vdash y : \texttt{Bool}}\,[\text{t-var}]}{\vdash \lambda y.y : t}\,[\text{t-lam}]}{\vdash (\lambda x.x)\,(\lambda y.y) : t}\,[\text{t-app}] \qquad \dfrac{}{\vdash \texttt{True} : \texttt{Bool}}\,[\text{t-bool}]}{\vdash (\lambda x.x)\,(\lambda y.y)\,\texttt{True} : \texttt{Bool}}\,[\text{t-app}]$$

Dove

- $t \stackrel{\text{def}}{=} \texttt{Bool} \to \texttt{Bool}$
- $s \stackrel{\text{def}}{=} t \to t = (\texttt{Bool} \to \texttt{Bool}) \to \texttt{Bool} \to \texttt{Bool}$

esempio

$\lambda x.\lambda y.\texttt{if}\ x\ y\ \texttt{False}$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash x : \texttt{Bool}}\,[\text{t-var}] \qquad \dfrac{}{\Gamma \vdash y : \texttt{Bool}}\,[\text{t-var}] \qquad \dfrac{}{\Gamma \vdash \texttt{False} : \texttt{Bool}}\,[\text{t-bool}]}{\Gamma \vdash \texttt{if}\ x\ y\ \texttt{False} : \texttt{Bool}}\,[\text{t-if}]}{x : \texttt{Bool} \vdash \lambda y.\texttt{if}\ x\ y\ \texttt{False} : \texttt{Bool} \to \texttt{Bool}}\,[\text{t-lam}]}{\vdash \lambda x.\lambda y.\texttt{if}\ x\ y\ \texttt{False} : \texttt{Bool} \to \texttt{Bool} \to \texttt{Bool}}\,[\text{t-lam}]$$

Dove

- $\Gamma \stackrel{\text{def}}{=} x : \texttt{Bool}, y : \texttt{Bool}$

esempio $\lambda x.\lambda y.\lambda z.z\,x\,y$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash z : t_1 \to t_2 \to s}\,[\text{t-var}] \qquad \dfrac{}{\Gamma \vdash x : t_1}\,[\text{t-var}]}{\Gamma \vdash z\,x : t_2 \to s}\,[\text{t-app}] \qquad \dfrac{}{\Gamma \vdash y : t_2}\,[\text{t-var}]}{x : t_1, y : t_2, z : t_1 \to t_2 \to s \vdash z\,x\,y : s}\,[\text{t-app}]}{x : t_1, y : t_2 \vdash \lambda z.z\,x\,y : (t_1 \to t_2 \to s) \to s}\,[\text{t-lam}]}{x : t_1 \vdash \lambda y.\lambda z.z\,x\,y : t_2 \to (t_1 \to t_2 \to s) \to s}\,[\text{t-lam}]}{\vdash \lambda x.\lambda y.\lambda z.z\,x\,y : t_1 \to t_2 \to (t_1 \to t_2 \to s) \to s}\,[\text{t-lam}]
$$

Dove

- $\Gamma \stackrel{\text{def}}{=} x : t_1, y : t_2, z : t_1 \to t_2 \to s$
- t_1, t_2 ed s sono tipi arbitrari

esercizi

Determinare quali delle seguenti espressioni sono ben tipate, cercando di costruire per ciascuna un albero di prova.

1. $\lambda f.\lambda x.f\ (f\ x)$
2. $\lambda x.x\ x$ No Autoapplicazione non è tipabile
3. `if True` $(\lambda x.\lambda y.x)\ (\lambda x.\lambda y.y)$
4. `if True` $(\lambda x.x)\ (\lambda x.\lambda y.y)$ No? In realtà si
5. $((\lambda x.x)$ `True`$)$ `False`
6. $(\lambda x.\lambda y.\lambda z.z\ x\ y)\ (\lambda x.x)$ `True`

Nota

- è possibile verificare le risposte chiedendo a `GHCi` il tipo di queste espressioni dimostrazioni (appendice facoltativa)

Teorema (progresso)

Se $\vdash M : t$ e $M \Rightarrow N \not\rightarrow$ allora N è un valore.

Dal lemma di subject reduction deduciamo $\vdash N : t$. Dimostriamo che $N \not\rightarrow$ implica che N è un valore per induzione su N e per casi sulla sua forma.

- Il caso $N = x$ è impossibile perché N è ben tipato nel contesto vuoto.
- Se $N = \mathtt{c}$ oppure $N = \lambda x.M'$ allora N è un valore e abbiamo finito.
- Caso $N = N_1\ N_2$:
  - deduciamo $\vdash N_1 : s \rightarrow t$ e $\vdash N_2 : s$ — da $\vdash N : t$ e [t-app]
  - deduciamo che $N_1 \not\rightarrow$ — da def. di $\rightarrow$
  - deduciamo che N_1 è un valore — ip. induttiva su N_1
  - deduciamo che N_1 è un'astrazione ed N è un redex — da $\vdash N_1 : s \rightarrow t$
  - questo caso è impossibile — da $N \not\rightarrow$
- Caso $N = \mathtt{if}\ N_1\ N_2\ N_3$:
  - deduciamo $\vdash N_1 : \mathtt{Bool}$ e $\vdash N_2 : t$ e $\vdash N_3 : t$ — da $\vdash N : t$ e [t-if]
  - deduciamo che $N_1 \not\rightarrow$ — da def. di $\rightarrow$
  - deduciamo che N_1 è un valore — ip. induttiva su N_1
  - deduciamo che N_1 è una costante ed N è un redex — da $\vdash N_1 : \mathtt{Bool}$
  - questo caso è impossibile — da $N \not\rightarrow$

Lemma (sostituzione)

Se $\Gamma, x : t \vdash M : s$ *e* $\Gamma \vdash N : t$ *allora* $\Gamma \vdash M\{N/x\} : s$*.*

Si procede per induzione su M e per casi sulla sua forma.

- Casi in cui $x \notin fv(M)$ e $M\{N/x\} = M$:
  - concludiamo $\Gamma \vdash M : t$ — rimozione delle ipotesi inutili
- Caso $M = x$ in cui $M\{N/x\} = N$ e $t = s$:
  - concludiamo $\Gamma \vdash N : t$ — ipotesi
- Caso $M = M_1\ M_2$ in cui $M\{N/x\} = M_1\{N/x\}\ M_2\{N/x\}$:
  - deduciamo $\Gamma, x : t \vdash M_1 : t' \to s$ e $\Gamma, x : t \vdash M_2 : t'$ — da [t-app]
  - deduciamo $\Gamma \vdash M_1\{N/x\} : t' \to s$ e $\Gamma \vdash M_2\{N/x\} : t'$ — ip. induttiva
  - concludiamo $\Gamma \vdash M\{N/x\} : s$ — usando [t-app]
- Caso $M = \lambda y.M'$. Possiamo assumere $x \neq y$ e $y \notin fv(N)$ grazie all'α-conversione, dunque $M\{N/x\} = \lambda y.M'\{N/x\}$. Ora:
  - deduciamo $\Gamma, x : t, y : t' \vdash M' : s'$ e $s = t' \to s'$ — da [t-lam]
  - otteniamo $\Gamma, y : t' \vdash N : t$ — lemma di indebolimento e $y \notin fv(N)$
  - deduciamo $\Gamma, y : t' \vdash M'\{N/x\} : s'$ — ip. induttiva
  - concludiamo $\Gamma \vdash M\{N/x\} : t$ — usando [t-lam]
- Il caso $M = \texttt{if}\ M_1\ M_2\ M_3$ è lasciato come esercizio.

Lemma (subject reduction)

Se $\Gamma \vdash M : t$ *e* $M \to N$ *allora* $\Gamma \vdash N : t$*.*

Si procede per induzione sulla derivazione di $M \to N$ e per casi sull'ultima regola applicata (si veda la def. di $\to$).

- Caso $M \to_\beta N$ in cui $M = (\lambda x.M')\, N'$ e $N = M'\{N'/x\}$:
  - deduciamo $\Gamma \vdash \lambda x.M' : s \to t$ e $\Gamma \vdash N' : s$ — da [t-app]
  - deduciamo $\Gamma, x : s \vdash M' : t$ — da [t-lam]
  - concludiamo $\Gamma \vdash N : t$ — lemma di sostituzione
- Caso $M \to_\eta N$ in cui $M = \lambda x.N\, x$ dove $x \notin fv(N)$:
  - deduciamo $\Gamma, x : t' \vdash N\, x : s$ e $t = t' \to s$ — da [t-lam]
  - deduciamo $\Gamma, x : t' \vdash N : s' \to s$ e $\Gamma, x : t' \vdash x : s'$ — da [t-app]
  - deduciamo $t' = s'$ — da [t-var]
  - concludiamo $\Gamma \vdash N : t$ — rimozione delle ipotesi inutili
- Caso $M = M_1\, M_2$ in cui $M_1 \to N_1$ e $N = N_1\, M_2$:
  - deduciamo $\Gamma \vdash M_1 : s \to t$ e $\Gamma \vdash M_2 : s$ — da [t-app]
  - deduciamo $\Gamma \vdash N_1 : s \to t$ — ip. induttiva
  - concludiamo $\Gamma \vdash N : t$ — usando [t-app]
- I casi rimanenti sono lasciati come esercizio.