Linguaggi di Programmazione

Roberta Gori

Semantica denotazionale dei comandi - 6.1,6.2

Lambda notazione

Lambda notazione

Modo per descrivere funzioni senza assegnargli un nome

funzione anonima

$\lambda x.\ e$ x funge da parametro formale in e

denota una funzione che attende che un valore che viene sostituito a x in tutte occorrenze di x in e e poi valuta e

applicazione della funzione anonima

$e_1\ e_2$ e_2 è l'argomento passato alla funzione e_1

denota l'applicazione della funzione e_1 ad e_2

riduce il bisogno di parentesi $e_1(e_2)$

Definizione di funzione

$$f(x) \triangleq x^2 - 2 \cdot x + 5$$

$$\lambda x.\ (x^2 - 2 \cdot x + 5)$$

le parentesi non sono necessarie
sono aggiunte solo per chiarezza

Regole associative

$e_1 \; e_2 \; e_3$ si legge $(e_1 \; e_2) \; e_3$ l'applicazione e' associativa a sinistra

$\lambda x. \; \lambda y. \; \lambda z. \; e$ si legge $\lambda x. \; (\lambda y. \; (\lambda z. \; e))$ l'astrazione e' associativa a destra

Scoping

$$\lambda x.\ e$$

lo scope di x e' e

x non e' visibile fuori da e

come una variabile locale

Alpha-conversione

$\lambda x.\ (x^2 - 2 \cdot x + 5)$

i nomi dei parametri formali sono inessenziali:

$\lambda y.\ (y^2 - 2 \cdot y + 5)$

le due espressioni denotano la stessa funzione

$\lambda x.\ e \equiv \lambda y.\ (e[^y/_x])$ (sotto alcune condizioni su e, y)

capture-avoiding substitution

(formalizzeremo a breve il concetto)

Applicazione (beta rule)

$$(\lambda x.\ e)\ e_0$$ applicazione di una funzione

$$\equiv$$

$$e[{}^{e_0}/_{x}]$$ valutazione via sostituzione

capture-avoiding substitution

Esempio

$\lambda x.\ (x^2 - 2 \cdot x + 5)$ una funzione

$(\lambda x.\ (x^2 - 2 \cdot x + 5))\ 2$ la sua applicazione

$$\equiv$$

$2^2 - 2 \cdot 2 + 5 = 5$ la sua valutazione

Esempio

$\lambda x.\ \lambda y.\ (x^2 - 2 \cdot y + 5)$ una funzione

$(\lambda x.\ \lambda y.\ (x^2 - 2 \cdot y + 5))\ 2$ la sua applicazione

$$\equiv$$

$\lambda y.\ (2^2 - 2 \cdot y + 5)$ la sua valutazione

e' ancora una funzione!

Esempio

$\lambda f.\ \lambda x.\ (x^2 + f\ 1)$ una funzione

$(\lambda f.\ \lambda x.\ (x^2 + f\ 1))\ (\lambda y.\ (2 \cdot y))$ la sua applicazione

$\equiv$ (l'argomento e' una funzione!)

$\lambda x.\ (x^2 + (\lambda y.\ (2 \cdot y))\ 1)$ la sua valutazione

posso usare funzioni di ordine superiore come argomenti/ risultati

Esempio

$\lambda f.\ \lambda x.\ (x^2 + f\ 1)$ una funzione

$(\lambda f.\ \lambda x.\ (x^2 + f\ 1))\ (\lambda y.\ (2 \cdot y))\ \ 3$ la sua applicazione

$$\equiv$$

$\lambda x.\ (x^2 + (\lambda y.\ (2 \cdot y))\ 1)\ \ 3$ la sua valutazione e la sua applicazione

$$\equiv$$

$3^2 + (\lambda y.\ (2 \cdot y))\ 1$ la sua valutazione e la sua applicazione

$$\equiv$$

$3^2 + 2 \cdot 1 = 11$ la sua valutazione

Condizionale in Lambda Notazione

Aggiungiamo all'astrazione $\lambda x \,.\, e$ e all'applicazione di funzione $e_1(e_2)$ anche il condizionale

$e \rightarrow e_1, e_2$ if e then e_1 else e_2

Posso scrivere $\lambda x \,.\, x > 0 \rightarrow 1{,}0$

esempio $\min \triangleq \lambda x.\ \lambda y.\ x < y \rightarrow x, y$

Semantica denotazionale dei comandi

Semantica denotazionale

$$\mathscr{C} : Com \to (\Sigma \rightharpoonup \Sigma) \qquad \mathscr{C} : Com \to (\Sigma \to \Sigma_\perp)$$

Lifting

$$(\cdot)^* : (\Sigma \to \Sigma_\perp) \to (\Sigma_\perp \to \Sigma_\perp)$$

$$f : \Sigma \to \Sigma_\perp \qquad f^* : \Sigma_\perp \to \Sigma_\perp$$

$$f^*(x) = \begin{cases} \perp & \text{if } x = \perp \\ f(x) & \text{otherwise} \end{cases}$$

$$\mathscr{C}\,[\![\mathbf{skip}]\!]\,\sigma \stackrel{\text{def}}{=} \sigma$$

$$\mathscr{C}\,[\![x := a]\!]\,\sigma \stackrel{\text{def}}{=} \sigma[^{\mathscr{A}\,[\![a]\!]\sigma}/_x]$$

$$\mathscr{C}\,[\![c_0; c_1]\!]\,\sigma \stackrel{\text{def}}{=} \mathscr{C}\,[\![c_1]\!]^*\,(\mathscr{C}\,[\![c_0]\!]\,\sigma)$$

$$\mathscr{C}\,[\![\mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1]\!]\,\sigma \stackrel{\text{def}}{=} \mathscr{B}\,[\![b]\!]\,\sigma \to \mathscr{C}\,[\![c_0]\!]\,\sigma, \mathscr{C}\,[\![c_1]\!]\,\sigma$$

$$\mathscr{C}\,[\![\mathbf{while}\ b\ \mathbf{do}\ c]\!]\,\sigma \stackrel{\text{def}}{=} \ ?$$

Semantica den. (con.)

$$\mathscr{C}\llbracket \textbf{while } b \textbf{ do } c \rrbracket \sigma \stackrel{\text{def}}{=} \mathscr{B}\llbracket b \rrbracket \sigma \to \mathscr{C}\llbracket \textbf{while } b \textbf{ do } c \rrbracket^* (\mathscr{C}\llbracket c \rrbracket \sigma), \sigma$$

$$\mathscr{C}\llbracket \textbf{while } b \textbf{ do } c \rrbracket \stackrel{\text{def}}{=} \lambda\sigma. \mathscr{B}\llbracket b \rrbracket \sigma \to \mathscr{C}\llbracket \textbf{while } b \textbf{ do } c \rrbracket^* (\mathscr{C}\llbracket c \rrbracket \sigma), \sigma$$

$$\equiv$$

$$(\lambda\varphi.\ \lambda\sigma.\ \mathscr{B}\llbracket b \rrbracket \sigma \to \varphi^*(\mathscr{C}\llbracket c \rrbracket \sigma), \sigma)\ \mathscr{C}\llbracket \textbf{while } b \textbf{ do } c \rrbracket$$

$$\Gamma_{b,c} \stackrel{\text{def}}{=} \lambda\varphi.\ \lambda\sigma.\ \mathscr{B}\llbracket b \rrbracket \sigma \to \varphi^*(\mathscr{C}\llbracket c \rrbracket \sigma), \sigma$$

$$\mathscr{C}\llbracket \textbf{while } b \textbf{ do } c \rrbracket = \Gamma_{b,c}\, \mathscr{C}\llbracket \textbf{while } b \textbf{ do } c \rrbracket$$

$p = f(p)$ una equazione di punto fisso!

Semantica den. (con.)

$$\underbrace{\mathscr{C}\,[\![\textbf{while}\ b\ \textbf{do}\ c]\!]}_{\Sigma\to\Sigma_\perp} = \Gamma_{b,c}\,\underbrace{\mathscr{C}\,[\![\textbf{while}\ b\ \textbf{do}\ c]\!]}_{\Sigma\to\Sigma_\perp}$$

$$\mathscr{C} : Com \to (\Sigma \to \Sigma_\perp)$$

$$\Gamma_{b,c} \stackrel{\text{def}}{=} \underbrace{\lambda\varphi.\ \underbrace{\lambda\sigma.\ \underbrace{\mathscr{B}\,[\![b]\!]\,\sigma \to \varphi^*(\mathscr{C}\,[\![c]\!]\,\sigma),\sigma}_{\Sigma_\perp}}_{\Sigma\to\Sigma_\perp}}_{(\Sigma\to\Sigma_\perp)\to\Sigma\to\Sigma_\perp}$$

$$\varphi : \Sigma \to \Sigma_\perp$$

$$\varphi^* : \Sigma_\perp \to \Sigma_\perp$$

$$\mathscr{C}\,[\![c]\!]\,\sigma : \Sigma_\perp$$

$$\varphi^*(\mathscr{C}\,[\![c]\!]\,\sigma) : \Sigma_\perp$$

$$\Gamma_{b,c} : (\Sigma \to \Sigma_\perp) \to \Sigma \to \Sigma_\perp$$

funzioni parziali

$$\Sigma \rightharpoonup \Sigma$$

insieme di coppie

$$(\sigma, \sigma')$$

$$\mathrm{OPC}_\perp$$

Monotono e continuo

$$\Gamma_{b,c} \stackrel{\text{def}}{=} \lambda\varphi.\ \lambda\sigma.\ \mathscr{B}\,[\![b]\!]\,\sigma \to \varphi^*(\mathscr{C}\,[\![c]\!]\,\sigma), \sigma$$

Prendiamo

$$R_{b,c} = \left\{ \frac{(\sigma'', \sigma')}{(\sigma, \sigma')} \mathcal{B}[\![b]\!]\sigma \wedge \mathcal{C}[\![c]\!]\sigma = \sigma''\ ,\ \ \frac{}{(\sigma, \sigma)} \mathcal{B}[\![\neg b]\!]\sigma \right\}$$

chiaramente

$\widehat{R}_{b,c} = \Gamma_{b,c}$ quando vediamo $\Gamma_{b,c}$ operare sulle funzioni parziali

$\widehat{R}_{b,c}$ e' (monotona e) continua, e cosi' anche $\Gamma_{b,c}$

$$\mathscr{C}\,[\![\textbf{while } b \textbf{ do } c]\!] \stackrel{\text{def}}{=} \text{fix } \Gamma_{b,c} = \bigsqcup_{n \in \mathbb{N}} \Gamma^n_{b,c}(\bot_{\Sigma \to \Sigma_\bot})$$

|

$\lambda\sigma.\ \bot$

Bottom

$\Sigma_\perp$ ha un elemento bottom: $\perp$

$\Sigma \to \Sigma_\perp$ ha un elemento bottom: $\lambda\sigma.\perp$

per evitare ambiguita'

denotiamo l'elemento bottom del dominio D con $\perp_D$

$\perp_{\Sigma_\perp}$ $\qquad\qquad$ $\perp_{\Sigma\to\Sigma_\perp}$

Esempio

$$w = \textbf{while true do skip}$$

$$\begin{aligned}\Gamma_{\mathbf{true},\mathbf{skip}}\varphi\sigma &= \mathscr{B}[\![\mathbf{true}]\!]\,\sigma \to \varphi^*\left(\mathscr{C}[\![\mathbf{skip}]\!]\,\sigma\right), \sigma \\ &= \mathbf{true} \to \varphi^*\left(\mathscr{C}[\![\mathbf{skip}]\!]\,\sigma\right), \sigma \\ &= \varphi^*\left(\mathscr{C}[\![\mathbf{skip}]\!]\,\sigma\right) \\ &= \varphi^*\sigma \\ &= \varphi\sigma\end{aligned}$$

$$\Gamma_{\mathbf{true},\mathbf{skip}}\varphi = \varphi$$

$\Gamma_{\mathbf{true},\mathbf{skip}}$ e' la funzione identita'
ogni elemento e'
un punto fisso

$$\mathrm{fix}\ \Gamma_{\mathbf{true},\mathbf{skip}} = \lambda\sigma.\ \perp_{\Sigma_\perp}$$

Esempio

$$w \triangleq \textbf{while } \underbrace{x > 1}_{b} \textbf{ do } \underbrace{x := x - 1}_{c}$$

$$\begin{aligned}\Gamma_{b,c}\ \varphi\ \sigma &= \mathcal{B}[\![x > 1]\!]\sigma \rightarrow \varphi^*(\mathcal{C}[\![x := x - 1]\!]\sigma), \sigma \\ &= (\sigma(x) > 1) \rightarrow \varphi^*(\sigma[^{\sigma(x)-1}/_{x}]), \sigma\end{aligned}$$

$$\widehat{R}_{b,c} \triangleq \left\{ \frac{}{(\sigma,\sigma)}\ \sigma(x) \leq 1 \quad , \quad \frac{(\sigma'',\sigma')}{(\sigma,\sigma')}\ \sigma(x) > 1 \wedge \sigma'' = \sigma[^{\sigma(x)-1}/_{x}] \right\}$$

$$\widehat{R}_{b,c} \triangleq \left\{ \frac{}{(\sigma,\sigma)}\ \sigma(x) \leq 1 \quad , \quad \frac{(\sigma[^{\sigma(x)-1}/_{x}],\sigma')}{(\sigma,\sigma')}\ \sigma(x) > 1 \right\}$$

Esempio

$$w \triangleq \textbf{while } x > 1 \textbf{ do } x := x - 1$$

$$\widehat{R}_{b,c} \triangleq \left\{ \frac{}{(\sigma,\sigma)}\ \sigma(x) \leq 1 \quad , \quad \frac{(\sigma[^{\sigma(x)-1}/_x], \sigma')}{(\sigma, \sigma')}\ \sigma(x) > 1 \right\}$$

$$\widehat{R}^{0}_{b,c}(\varnothing) = \varnothing$$

$$\widehat{R}^{1}_{b,c}(\varnothing) = \{(\sigma,\sigma) \mid \sigma(x) \leq 1\} \qquad \boxed{\supseteq \{(\sigma,\sigma) \mid \sigma(x) = 1\}}$$

$$\widehat{R}^{2}_{b,c}(\varnothing) = \widehat{R}^{1}_{b,c}(\varnothing) \cup \{(\sigma, \sigma[1/x]) \mid \sigma(x) = 2\}$$

$$\boxed{\supseteq \{(\sigma, \sigma[1/x]) \mid \sigma(x) = 2\}}$$

$$\widehat{R}^{3}_{b,c}(\varnothing) = \widehat{R}^{2}_{b,c}(\varnothing) \cup \{(\sigma, \sigma[1/x]) \mid \sigma(x) = 3\}$$

$$\dots$$

$$\widehat{R}^{n}_{b,c}(\varnothing) = \{(\sigma,\sigma) \mid \sigma(x) \leq 1\} \cup \{(\sigma, \sigma[1/x]) \mid 1 < \sigma(x) \leq n\}$$

$$\dots$$

$$\sqcup_n \widehat{R}^{n}_{b,c}(\varnothing) = \{(\sigma,\sigma) \mid \sigma(x) \leq 1\} \cup \{(\sigma, \sigma[1/x]) \mid 1 < \sigma(x)\}$$