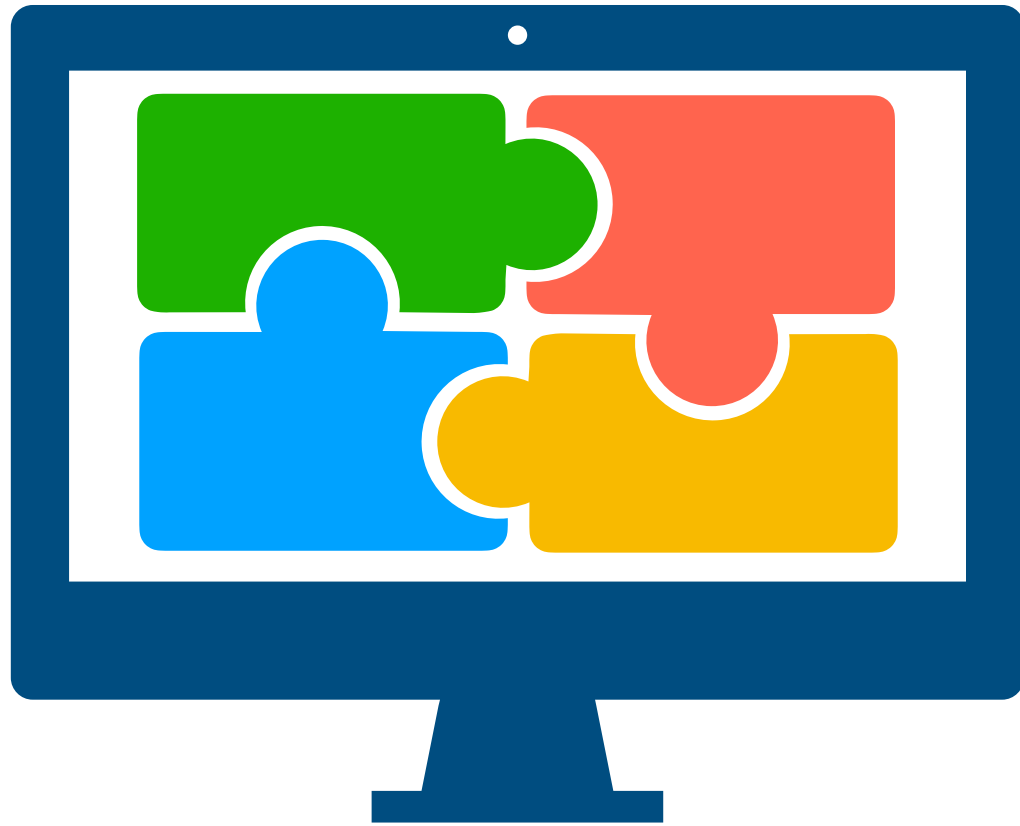




Linguaggi di Programmazione

Roberta Gori

Semantica denotazionale dei comandi - 6.1,6.2

Lambda notazione

Lambda notazione

Ingredienti principali

funzioni anonime

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

denota una funzione che attende che un valore sia sostituito a x e poi valuta e

applicazione

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

denota l'applicazione della funzione e_1 a e_2

riduce il bisogno di parentesi $e_1(e_2)$

Definizione di funzione

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

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


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 è e

x non è 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

$\equiv$

$$\lambda x. (x^2 + (\lambda y. (2 \cdot y)) \ 1) \quad 3$$

la sua valutazione
la sua applicazione

$\equiv$

$$3^2 + (\lambda y. (2 \cdot y)) \ 1$$

la sua valutazione
la sua applicazione

$\equiv$

$$3^2 + 2 \cdot 1 = 11$$

la sua valutazione

Condizionale

$e \rightarrow e_1, e_2$

if e then e_1 else e_2

esempio

$\text{min} \triangleq \lambda x. \lambda y. x < y \rightarrow x, y$

Semantica denotazionale dei comandi

Semantica denotazionale

$$\mathcal{C} : Com \rightarrow (\Sigma \rightarrow \Sigma)$$

$$\mathcal{C} : Com \rightarrow (\Sigma \rightarrow \Sigma_{\perp})$$

$$\mathcal{C} [\mathbf{skip}] \sigma \stackrel{\text{def}}{=} \sigma$$

$$\mathcal{C} [x := a] \sigma \stackrel{\text{def}}{=} \sigma[\mathcal{A} [a] \sigma / x]$$

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

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

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

Lifting

$$(\cdot)^* : (\Sigma \rightarrow \Sigma_{\perp}) \rightarrow (\Sigma_{\perp} \rightarrow \Sigma_{\perp})$$

$$f : \Sigma \rightarrow \Sigma_{\perp} \quad f^* : \Sigma_{\perp} \rightarrow \Sigma_{\perp}$$

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

Semantica den. (con.)

$$\begin{aligned}\mathcal{C} \llbracket \mathbf{while} \ b \ \mathbf{do} \ c \rrbracket \sigma &\stackrel{\text{def}}{=} \mathcal{B} \llbracket b \rrbracket \sigma \rightarrow \mathcal{C} \llbracket \mathbf{while} \ b \ \mathbf{do} \ c \rrbracket^* (\mathcal{C} \llbracket c \rrbracket \sigma), \sigma \\ \mathcal{C} \llbracket \mathbf{while} \ b \ \mathbf{do} \ c \rrbracket &\stackrel{\text{def}}{=} \lambda \sigma. \mathcal{B} \llbracket b \rrbracket \sigma \rightarrow \mathcal{C} \llbracket \mathbf{while} \ b \ \mathbf{do} \ c \rrbracket^* (\mathcal{C} \llbracket c \rrbracket \sigma), \sigma \\ &\equiv \\ &(\lambda \varphi. \lambda \sigma. \mathcal{B} \llbracket b \rrbracket \sigma \rightarrow \varphi^* (\mathcal{C} \llbracket c \rrbracket \sigma), \sigma) \ \mathcal{C} \llbracket \mathbf{while} \ b \ \mathbf{do} \ c \rrbracket\end{aligned}$$

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

$$\mathcal{C} \llbracket \mathbf{while} \ b \ \mathbf{do} \ c \rrbracket = \Gamma_{b,c} \ \mathcal{C} \llbracket \mathbf{while} \ b \ \mathbf{do} \ c \rrbracket$$

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

Semantica den. (con.)

$$\underbrace{\mathcal{C} [\mathbf{while} \ b \ \mathbf{do} \ c]}_{\Sigma \rightarrow \Sigma_{\perp}} = \Gamma_{b,c} \underbrace{\mathcal{C} [\mathbf{while} \ b \ \mathbf{do} \ c]}_{\Sigma \rightarrow \Sigma_{\perp}}$$

$$\mathcal{C} : Com \rightarrow (\Sigma \rightarrow \Sigma_{\perp})$$

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

$$\varphi : \Sigma \rightarrow \Sigma_{\perp}$$

$$\varphi^* : \Sigma_{\perp} \rightarrow \Sigma_{\perp}$$

$$\mathcal{C} [c] \sigma : \Sigma_{\perp}$$

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

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

funzioni parziali

$$\Sigma \multimap \Sigma$$

insieme di coppie

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

OPC_⊥

Monotono e continuo

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

Prendiamo

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

chiaramente

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

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

$$\mathcal{C} \llbracket \mathbf{while} \ b \ \mathbf{do} \ c \rrbracket \stackrel{\text{def}}{=} \text{fix } \Gamma_{b,c} = \bigsqcup_{n \in \mathbb{N}} \Gamma_{b,c}^n(\perp_{\Sigma \rightarrow \Sigma_{\perp}})$$

Bottom

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

$\Sigma \rightarrow \Sigma_{\perp}$ ha un elemento bottom: $\lambda\sigma. \perp$

per evitare ambiguità'

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

$\perp_{\Sigma_{\perp}}$

$\perp_{\Sigma \rightarrow \Sigma_{\perp}}$

Esempio

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

$$\begin{aligned}\Gamma_{\text{true,skip}} \varphi \sigma &= \mathcal{B} \llbracket \text{true} \rrbracket \sigma \rightarrow \varphi^* (\mathcal{C} \llbracket \text{skip} \rrbracket \sigma), \sigma \\ &= \text{true} \rightarrow \varphi^* (\mathcal{C} \llbracket \text{skip} \rrbracket \sigma), \sigma \\ &= \varphi^* (\mathcal{C} \llbracket \text{skip} \rrbracket \sigma) \\ &= \varphi^* \sigma \\ &= \varphi \sigma\end{aligned}$$

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

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

Esempio

$$w \triangleq \text{while } \underbrace{x > 1}_b \text{ do } \underbrace{x := x - 1}_c$$

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

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

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

Esempio

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

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

$$\hat{R}_{b,c}^0(\emptyset) = \emptyset$$

$$\hat{R}_{b,c}^1(\emptyset) = \{(\sigma, \sigma) \mid \sigma(x) \leq 1\} \supseteq (\sigma, \sigma) \text{ tale che } \sigma(x) = 1$$

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

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

...

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

...

$$\mathcal{C}[[w]] = \text{fix}(\hat{R}_{b,c}) = \{(\sigma, \sigma) \mid \sigma(x) \leq 1\} \cup \{(\sigma, \sigma[1/x]) \mid 1 < \sigma(x)\}$$