PSC 2023/24 (375AA, 9CFU)

Principles for Software Composition

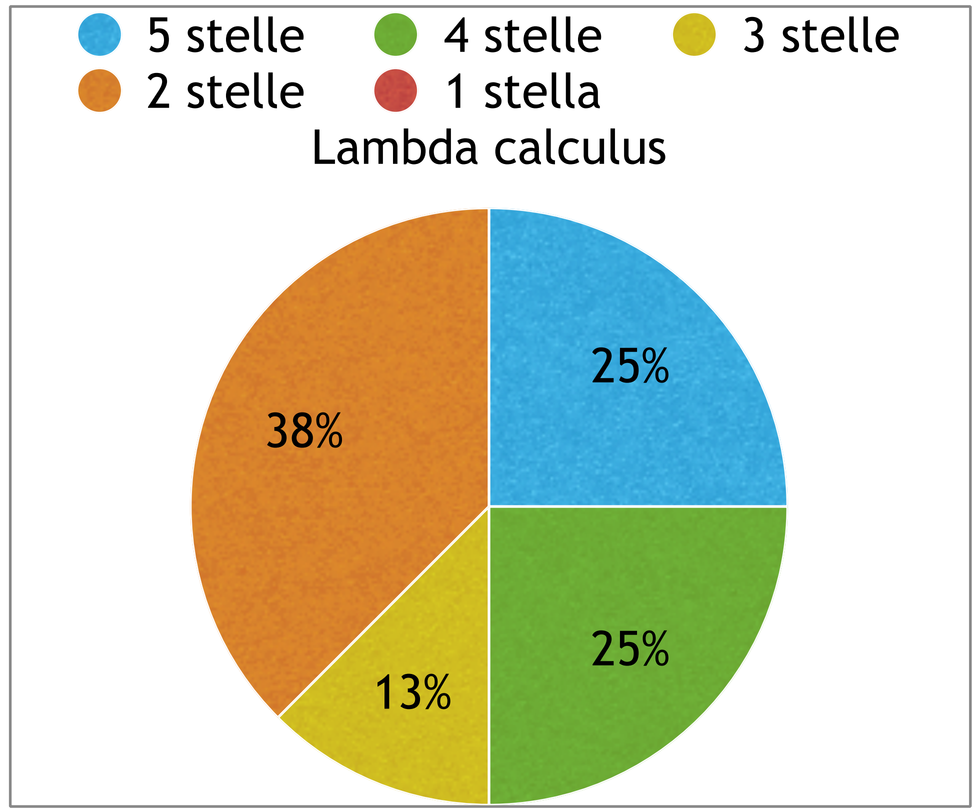
Roberto Bruni
http://www.di.unipi.it/~bruni/

http://didawiki.di.unipi.it/doku.php/magistraleinformatica/psc/start

09 - Denotational semantics of commands

Lambda notation

From your forms



(over 8 answers)

Lambda notation

Key ingredients

anonymous functions

$\lambda x.\ e$ x serves as a formal parameter in e

denotes a function that waits for one value to be substituted for x and then evaluates e

application

$e_1\ e_2$ e_2 is the argument passed to the function e_1

denotes the application of the function e_1 to e_2

reduces the need of parentheses $e_1(e_2)$

Function definition

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

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

unnecessary parentheses
added for clarity

Associative rules

$e_1 \; e_2 \; e_3$	is read	$(e_1 \; e_2) \; e_3$	application is left-associative
$\lambda x. \; \lambda y. \; \lambda z. \; e$	is read	$\lambda x. \; (\lambda y. \; (\lambda z. \; e))$	abstraction is right-associative

Scoping

$\lambda x.\ e$

the scope of x is e

x not visible outside e

like a local variable

Alpha-conversion

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

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

names of formal parameters are inessential: the two expressions denote the same function

$\lambda x.\ e \equiv \lambda y.\ (e[{}^{y}/_{x}])$ (under suitable conditions on e, y)

capture-avoiding substitution (to be formalised later)

Application (beta rule)

$$(\lambda x.\ e)\ e_0$$

application of a function

$$\equiv$$

$$e[^{e_0}/_x]$$

evaluation via substitution

capture-avoiding substitution

Example

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

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

$\equiv$

$2^2 - 2 \cdot 2 + 5 = 5$ its evaluation

Example

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

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

$\equiv$

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

it is still a function!

Example

$\lambda f.\ \lambda x.\ (x^2 + f\ 1)$ a function

$(\lambda f.\ \lambda x.\ (x^2 + f\ 1))\ (\lambda y.\ (2 \cdot y))$ its application

$\equiv$ (the argument is a function!)

$\lambda x.\ (x^2 + (\lambda y.\ (2 \cdot y))\ 1)$ its evaluation

higher-order: functions as arguments/results

Example

$\lambda f.\ \lambda x.\ (x^2 + f\ 1)$	a function
$(\lambda f.\ \lambda x.\ (x^2 + f\ 1))\ (\lambda y.\ (2 \cdot y))\ \ 3$	its application
$\equiv$	
$\lambda x.\ (x^2 + (\lambda y.\ (2 \cdot y))\ 1)\ \ \ \ 3$	its evaluation its application
$\equiv$	
$3^2 + (\lambda y.\ (2 \cdot y))\ 1$	its evaluation its application
$\equiv$	
$3^2 + 2 \cdot 1 = 11$	its evaluation

Conditional

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

example $\mathrm{min} \triangleq \lambda x.\ \lambda y.\ x < y \to x, y$

From recursion to fixpoint

$$fact\ n = (n < 2) \to 1\,, n \cdot fact(n-1)$$

$$fact = \lambda n\,.\,(n < 2) \to 1\,, n \cdot fact(n-1)$$

$$fact = (\lambda f\,.\,\lambda n\,.\,(n < 2) \to 1\,, n \cdot f(n-1))\ fact$$

$$\Gamma = \lambda f\,.\,\lambda n\,.\,(n < 2) \to 1\,, n \cdot f(n-1)$$

$$fact = \Gamma(fact)$$

$$fact = \mathsf{fix}\ \Gamma$$

From recursion to fixpoint

$$\Gamma = \lambda f\,.\,\lambda n\,.\,(n < 2) \rightarrow 1\,, n \cdot f(n-1)$$

$$id = \lambda x\,.\,x$$

$$\begin{aligned}\Gamma\ id &= (\lambda f\,.\,\lambda n\,.\,(n < 2) \rightarrow 1\,, n \cdot f(n-1))\ id \\ &= \lambda n\,.\,(n < 2) \rightarrow 1\,, n \cdot id(n-1) \\ &= \lambda n\,.\,(n < 2) \rightarrow 1\,, n \cdot (n-1) \\ &\neq id\end{aligned}$$

From recursion to fixpoint

$$\Gamma = \lambda f \,.\, \lambda n \,.\, (n < 2) \rightarrow 1 \,,\, n \cdot f(n-1)$$

$$succ = \lambda x \,.\, x + 1$$

$$\begin{aligned} \Gamma\ succ &= (\lambda f \,.\, \lambda n \,.\, (n < 2) \rightarrow 1 \,,\, n \cdot f(n-1))\ succ \\ &= \lambda n \,.\, (n < 2) \rightarrow 1 \,,\, n \cdot succ(n-1) \\ &= \lambda n \,.\, (n < 2) \rightarrow 1 \,,\, n \cdot n \\ &\neq succ \end{aligned}$$

From recursion to fixpoint

$$\Gamma = \lambda f\,.\,\lambda n\,.\,(n < 2) \rightarrow 1\,,\,n \cdot f(n-1)$$

$$square = \lambda x\,.\,x^2$$

$$\begin{aligned}\Gamma\ square &= (\lambda f\,.\,\lambda n\,.\,(n < 2) \rightarrow 1\,,\,n \cdot f(n-1))\ square \\ &= \lambda n\,.\,(n < 2) \rightarrow 1\,,\,n \cdot square(n-1) \\ &= \lambda n\,.\,(n < 2) \rightarrow 1\,,\,n \cdot (n-1)^2 \\ &\neq square\end{aligned}$$

From recursion to fixpoint

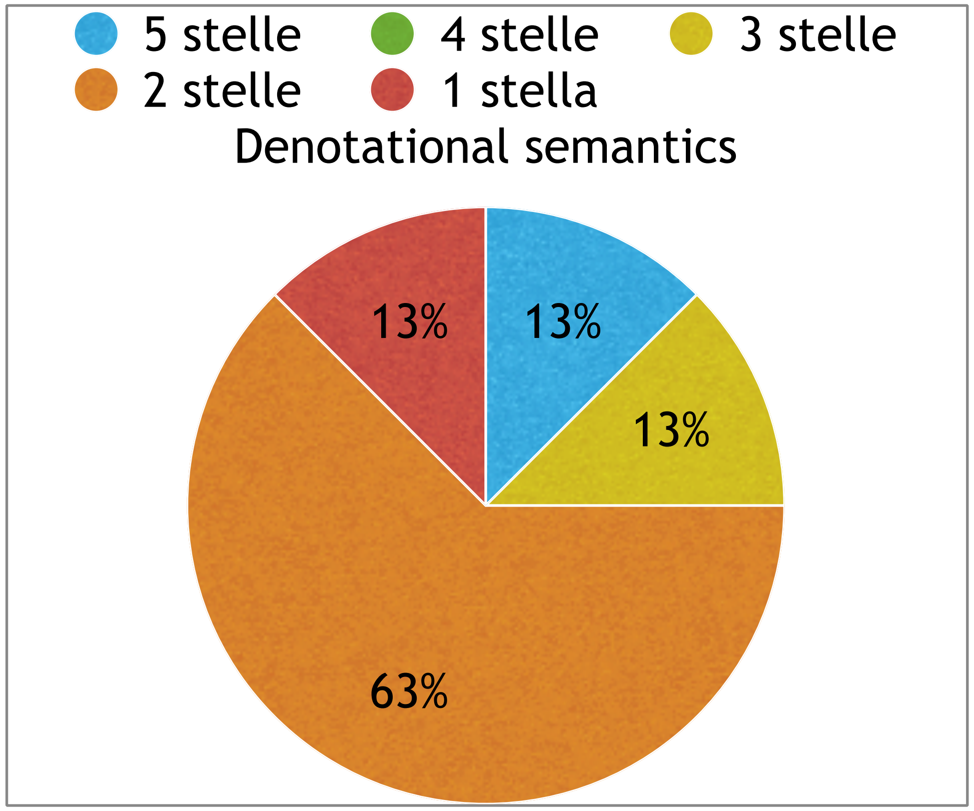
$$\Gamma = \lambda f\,.\,\lambda n\,.\,(n < 2) \rightarrow 1\,,\,n \cdot f(n-1)$$

$$fact = \lambda x\,.\,x\,!$$

$$\begin{aligned}\Gamma\; fact &= (\lambda f\,.\,\lambda n\,.\,(n < 2) \rightarrow 1\,,\,n \cdot f(n-1))\; fact \\ &= \lambda n\,.\,(n < 2) \rightarrow 1\,,\,n \cdot fact(n-1) \\ &= \lambda n\,.\,(n < 2) \rightarrow 1\,,\,n \cdot (n-1)\,! \\ &= fact\end{aligned}$$

Denotational semantics of commands

From your forms



(over 8 answers)

Denotational semantics

$$\mathscr{C} : Com \rightarrow (\Sigma \rightharpoonup \Sigma)$$

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

Lifting

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

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

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

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

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

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

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

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

Denotational sem. (ctd)

$$\mathscr{C}\,[\![\textbf{while } b \textbf{ do } c]\!]\,\sigma \stackrel{\text{def}}{=} \mathscr{B}\,[\![b]\!]\,\sigma \rightarrow \mathscr{C}\,[\![\textbf{while } b \textbf{ do } c]\!]^*\,(\mathscr{C}\,[\![c]\!]\,\sigma)\,,\sigma$$

$$\mathscr{C}\,[\![\textbf{while } b \textbf{ do } c]\!] \stackrel{\text{def}}{=} \lambda\sigma.\mathscr{B}\,[\![b]\!]\,\sigma \rightarrow \mathscr{C}\,[\![\textbf{while } b \textbf{ do } c]\!]^*\,(\mathscr{C}\,[\![c]\!]\,\sigma)\,,\sigma$$

$$\equiv$$

$$(\lambda\varphi.\ \lambda\sigma.\ \mathscr{B}\,[\![b]\!]\,\sigma \rightarrow \varphi^*(\mathscr{C}\,[\![c]\!]\,\sigma),\sigma)\ \ \mathscr{C}\,[\![\textbf{while } b \textbf{ do } c]\!]$$

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

$$\mathscr{C}\,[\![\textbf{while } b \textbf{ do } c]\!] = \Gamma_{b,c}\,\mathscr{C}\,[\![\textbf{while } b \textbf{ do } c]\!]$$

$p = f(p)$ a fixpoint equation!

Denotational sem. (ctd)

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

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

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

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

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

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

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

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

partial functions

$\Sigma \rightharpoonup \Sigma$

sets of pairs

(σ, σ')

$\mathrm{CPO}_\perp$

Monotone and continuous

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

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

clearly $\widehat{R}_{b,c} = \Gamma_{b,c}$ when we see $\Gamma_{b,c}$ as operating over partial functions

$\widehat{R}_{b,c}$ is (monotone and) continuous, and so is $\Gamma_{b,c}$

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

$\lambda\sigma.\ \bot$

Bottom

$\Sigma_\perp$ has a bottom element: $\perp$

$\Sigma \to \Sigma_\perp$ has a bottom element: $\lambda\sigma.\ \perp$

to avoid ambiguities

we denote the bottom element of a domain D by $\perp_D$

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

Example

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

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

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

$\Gamma_{\textbf{true},\textbf{skip}}$ is the identity function every element is a fixpoint

$$\text{fix } \Gamma_{\textbf{true},\textbf{skip}} = \lambda\sigma.\ \bot_{\Sigma_\bot}$$

Example

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

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

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

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

Example

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

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

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

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

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

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

$$\dots$$

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

$$\dots$$

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