~announce past pape~

~min sheet~

~last Sam lecture~

- guest lectures!
- office hours available until Dec

Call-by-name Vs. Call-by-value

Call-by-name (CBN)

= where variables represent terms

Example:

$f(x) = x + x$

$\{x = 1+1\}$

$f(1+1)$

$\to (1+1) + (1+1)$

$\to 2 + (1+1)$ 4 eval steps

$\to 2+2$

$\to 4$

call-by-value (CBV)

= where variables represent values

$f(1+1)$

$\to f(2)$

$\to 2+2$ 3 eval steps

$\to 4$

Evaluation order does not affect pure results

The difference is only noticable in the presence of effects

printing, errors ...

CBV languages:

- C
- Java
- Scala
- JS
- OCaml
- scheme
- ...

Haskell = call-by-need

optimised variant of CB name

Call-by-value λ-calc

Reference: Simply-typed λ-calculus

Alex Kavvos

No need to change statics

Figure 1: Statics of the simply-typed λ-calculus (with numbers)

$$\frac{}{\Gamma, x : \sigma \vdash x : \sigma}\ \textsc{Var} \qquad \frac{n \in \mathbb{N}}{\Gamma \vdash \mathsf{num}[n] : \mathsf{Num}}\ \textsc{Num} \qquad \frac{\Gamma \vdash e_1 : \mathsf{Num} \qquad \Gamma \vdash e_2 : \mathsf{Num}}{\Gamma \vdash \mathsf{plus}(e_1; e_2) : \mathsf{Num}}\ \textsc{Plus}$$

$$\frac{\Gamma \vdash e_1 : \mathsf{Num} \qquad \Gamma \vdash e_2 : \mathsf{Num}}{\Gamma \vdash \mathsf{times}(e_1; e_2) : \mathsf{Num}}\ \textsc{Times} \qquad \frac{\Gamma \vdash e_1 : \sigma_1 \qquad \Gamma, x : \sigma_1 \vdash e_2 : \sigma_2}{\Gamma \vdash \mathsf{let}(e_1; x.\, e_2) : \sigma_2}\ \textsc{Let} \qquad \frac{}{\Gamma \vdash \langle\rangle : \mathbf{1}}\ \textsc{Unit}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash \langle e_1, e_2 \rangle : \tau_1 \times \tau_2}\ \textsc{Prod} \qquad \frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash \pi_1(e) : \tau_1}\ \textsc{Proj-1} \qquad \frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash \pi_2(e) : \tau_2}\ \textsc{Proj-2} \qquad \frac{\Gamma \vdash e : \mathbf{0}}{\Gamma \vdash \mathsf{abort}(e) : \tau}\ \textsc{Abort}$$

$$\frac{\Gamma \vdash e : \tau_1}{\Gamma \vdash \mathsf{inl}(e) : \tau_1 + \tau_2}\ \textsc{Inl} \qquad \frac{\Gamma \vdash e : \tau_2}{\Gamma \vdash \mathsf{inr}(e) : \tau_1 + \tau_2}\ \textsc{Inr}$$

$$\frac{\Gamma \vdash e : \tau_1 + \tau_2 \qquad \Gamma, x : \tau_1 \vdash e_1 : \tau \qquad \Gamma, y : \tau_2 \vdash e_2 : \tau}{\Gamma \vdash \mathsf{case}(e; x.\, e_1; y.\, e_2) : \tau}\ \textsc{Case} \qquad \frac{\Gamma, x : \sigma \vdash e : \tau}{\Gamma \vdash \lambda x : \sigma.\, e : \sigma \to \tau}\ \textsc{Lam}$$

$$\frac{\Gamma \vdash e_1 : \sigma \to \tau \qquad \Gamma \vdash e_2 : \sigma}{\Gamma \vdash e_1(e_2) : \tau}\ \textsc{App}$$

eval order is dynamic.

the lecture notes actually ignore products when exploring CBV STLC, so this was just me missing expanding to that. I agree there should be e_1 val and e_2 val as there is for sums.

needed for passing into functions / lets

why no val here

required?

Yes both required

Figure 2: Dynamics of the simply-typed λ-calculus

$$\textsc{Val-Unit}\ \frac{}{\langle\rangle\ \mathsf{val}} \qquad \textsc{Val-Pair}\ \frac{}{\langle e_1, e_2\rangle\ \mathsf{val}} \qquad \textsc{Val-Inl}\ \frac{e\ \mathsf{val}}{\mathsf{inl}(e)\ \mathsf{val}} \qquad \textsc{Val-Inr}\ \frac{e\ \mathsf{val}}{\mathsf{inr}(e)\ \mathsf{val}} \qquad \textsc{Val-Lam}\ \frac{}{\lambda x:\tau.\,e\ \mathsf{val}}$$

$$\textsc{Val-Num}\ \frac{n \in \mathbb{N}}{\mathsf{num}[n]\ \mathsf{val}} \qquad \textsc{D-Plus}\ \frac{n_1 + n_2 = n}{\mathsf{plus}(\mathsf{num}[n_1];\mathsf{num}[n_2]) \longmapsto \mathsf{num}[n]} \qquad \textsc{D-Plus-1}\ \frac{e_1 \longmapsto e_1'}{\mathsf{plus}(e_1;e_2) \longmapsto \mathsf{plus}(e_1';e_2)}$$

$$\textsc{D-Plus-2}\ \frac{e_1\ \mathsf{val} \quad e_2 \longmapsto e_2'}{\mathsf{plus}(e_1;e_2) \longmapsto \mathsf{plus}(e_1;e_2')} \qquad \textsc{D-Let}\ \frac{e_1\ \mathsf{val}}{\mathsf{let}(e_1; x.\,e_2) \longmapsto e_2[e_1/x]} \qquad \textsc{D-Proj-Tuple-1}\ \frac{}{\pi_1(\langle e_1, e_2\rangle) \longmapsto e_1}$$

$$\textsc{D-Proj-Tuple-2}\ \frac{}{\pi_1(\langle e_1, e_2\rangle) \longmapsto e_2} \qquad \textsc{D-Proj-1}\ \frac{e \longmapsto e'}{\pi_1(e) \longmapsto \pi_1(e')} \qquad \textsc{D-Proj-2}\ \frac{e \longmapsto e'}{\pi_2(e) \longmapsto \pi_2(e')}$$

$$\textsc{D-Abort-1}\ \frac{e \longmapsto e'}{\mathsf{abort}(e) \longmapsto \mathsf{abort}(e')} \qquad \textsc{D-Case-Inl}\ \frac{e\ \mathsf{val}}{\mathsf{case}(\mathsf{inl}(e); x.\,e_1; y.\,e_2) \longmapsto e_1[e/x]}$$

needed to trigger full eval of e.

$$\textsc{D-Case-Inr}\ \frac{e\ \mathsf{val}}{\mathsf{case}(\mathsf{inr}(e); x.\,e_1; y.\,e_2) \longmapsto e_2[e/y]} \qquad \textsc{D-Case-1}\ \frac{e \longmapsto e'}{\mathsf{case}(e; x.\,e_1; y.\,e_2) \longmapsto \mathsf{case}(e'; x.\,e_1; y.\,e_2)}$$

$$\textsc{D-App-1}\ \frac{e_1 \longmapsto e_1'}{e_1(e_2) \longmapsto e_1'(e_2)} \qquad \textsc{D-Beta}\ \frac{e_2\ \mathsf{val}}{(\lambda x:\tau.\,e_1)(e_2) \longmapsto e_1[e_2/x]}$$

$$\text{D-INL}\ \frac{e \mapsto e'}{\mathrm{inl}(e) \to \mathrm{inl}(e')} \qquad \text{D-INR}\ \frac{e \mapsto e'}{\mathrm{inr}(e) \mapsto \mathrm{inr}(e')}$$

$$\text{D-APP-2}\ \frac{e_1\ \mathrm{val} \qquad e_2 \mapsto e_2'}{e_1(e_2) \mapsto e_1(e_2')}$$

D-LET ?

This is an error in the notes that has been missed for 4 years! good spot! → I will correct the latex notes

Figure 2: Dynamics of the simply-typed λ-calculus

Val-Unit
$$\frac{}{\langle\rangle\ \mathsf{val}}$$

Val-Pair
$$\frac{}{\langle e_1, e_2\rangle\ \mathsf{val}}$$

Val-Inl
$$\frac{}{\mathsf{inl}(e)\ \mathsf{val}}$$

Val-Inr
$$\frac{}{\mathsf{inr}(e)\ \mathsf{val}}$$

Val-Lam
$$\frac{}{\lambda x : \tau.\, e\ \mathsf{val}}$$

Val-Num
$$\frac{n \in \mathbb{N}}{\mathsf{num}[n]\ \mathsf{val}}$$

D-Plus
$$\frac{n_1 + n_2 = n}{\mathsf{plus}(\mathsf{num}[n_1]; \mathsf{num}[n_2]) \longmapsto \mathsf{num}[n]}$$

D-Plus-1
$$\frac{e_1 \longmapsto e_1'}{\mathsf{plus}(e_1; e_2) \longmapsto \mathsf{plus}(e_1'; e_2)}$$

D-Plus-2
$$\frac{e_1\ \mathsf{val} \qquad e_2 \longmapsto e_2'}{\mathsf{plus}(e_1; e_2) \longmapsto \mathsf{plus}(e_1; e_2')}$$

D-Let
$$\frac{}{\mathsf{let}(e_1; x.\, e_2) \longmapsto e_2[e_1/x]}$$

D-Proj-Tuple-1
$$\frac{}{\pi_1(\langle e_1, e_2\rangle) \longmapsto e_1}$$

D-Proj-Tuple-2
$$\frac{}{\pi_1(\langle e_1, e_2\rangle) \longmapsto e_2}$$

D-Proj-1
$$\frac{e \longmapsto e'}{\pi_1(e) \longmapsto \pi_1(e')}$$

D-Proj-2
$$\frac{e \longmapsto e'}{\pi_2(e) \longmapsto \pi_2(e')}$$

D-Abort-1
$$\frac{e \longmapsto e'}{\mathsf{abort}(e) \longmapsto \mathsf{abort}(e')}$$

D-Case-Inl
$$\frac{}{\mathsf{case}(\mathsf{inl}(e); x.\, e_1; y.\, e_2) \longmapsto e_1[e/x]}$$

D-Case-Inr
$$\frac{}{\mathsf{case}(\mathsf{inr}(e); x.\, e_1; y.\, e_2) \longmapsto e_2[e/y]}$$

D-Case-1
$$\frac{e \longmapsto e'}{\mathsf{case}(e; x.\, e_1; y.\, e_2) \longmapsto \mathsf{case}(e'; x.\, e_1; y.\, e_2)}$$

D-App-1
$$\frac{e_1 \longmapsto e_1'}{e_1(e_2) \longmapsto e_1'(e_2)}$$

D-Beta
$$\frac{}{(\lambda x : \tau.\, e_1)(e_2) \longmapsto e_1[e_2/x]}$$

Call-by-value λ-calc-cont

Progress + Preservation hold for CBV STLC.

Effects

Printing effect

Statics:

$$\text{PRINT}\ \frac{s \in \Sigma^{*} \quad \Gamma \vdash e : \tau}{\Gamma \vdash \text{print}(s; e) : \tau}$$

Dynamics:

$$e \overset{s}{\longmapsto} e'$$

e prints s and steps to e'
(ε is empty string)

$$\text{D-P-PRINT}\ \frac{}{\text{print}(s, e) \overset{s}{\longmapsto} e}$$

$$\text{D-P-BETA}\ \frac{(v \text{ val})}{(\lambda x : \tau . e)(v) \overset{\varepsilon}{\longmapsto} e[v/x]}$$

$$\text{D-P-APP-1}\ \frac{e_1 \overset{s}{\longmapsto} e_1'}{e_1(e_2) \overset{s}{\longmapsto} e_1'(e_2)}$$

Examples with Effects

$\vdash (\lambda x{:}Num.\,plus(x;x))(print(hi;num[1])) : Num$

CBN:

$(\lambda x{:}Num.\,plus(x;x))(print(hi;num[1]))$

$\mapsto_n$

$plus(print(hi;num[1]);\,print(hi;num[1]))$

$\xmapsto{hi}_n$

$plus(num[1];\,print(hi;num[1]))$

$\xmapsto{hi}_n$

$plus(num[1];\,num[1])$

$\mapsto_n$

$num[2]$

4 eval steps
"hi" ×2

$(\lambda x{:}Num.\,plus(x;x))(print(hi;num[1]))$

$\xmapsto{hi}_v$

$(\lambda x{:}Num.\,plus(x;x))\,(num[1])$

$\mapsto_v$

$plus(num[1];\,num[1])$

$\mapsto_v$

$num[2]$

3 eval steps
"hi" ×1

CBV vs CBN

- same pure output
- differences in eval steps
- different observationally in presence of effects

(CBPV - call-by-push-value)

~ remind + enthuse about guest lectures ~