CS 520

Theory of Programming Language

05/12 – 05/26, 2021

$\boxed{\text{Lambda Calculus} \cdots \text{Denotational Semantics.}}$

1. Reminder.

$$\langle exp \rangle ::= \langle var \rangle \mid \underline{\lambda \langle var \rangle . \langle exp \rangle} \mid \underline{\langle exp \rangle \langle exp \rangle}.$$

$\rightarrow$ $\cdots$ contraction, $\rightarrow^{*}$ $\cdots$ reduction $\cdots$ β-normal form.

$\Rightarrow_{normal}$ $\cdots$ normal-order evaluation. $\Rightarrow_{eager}$ $\cdots$ eager evaluation.

canonical form (values) $\cdots$ λ-abstractions ($\lambda v. e$).

2. Denotational semantics. closed expression.

D. ⋯ space $[\![e]\!] \in D$.

① 1960's ⋯ open problem. solved later by Dana Scott. via domain theory.

$D \simeq D \to D$.

② Pb. S ⋯ set.

$[S \to S] \subseteq S$ ⋯ (*)

what kind of S is possible? $|S|$ should be at most 1.

why? Because every fn f on S has a fixed point.

why?

Supp. $[S \to S] \subseteq S$.

Let $f \in [S \to S]$.

Define $p : S \to S$ by. $\underset{S}{\overset{\Uparrow}{p(x)}} = \begin{cases} f(x(x)) & \text{if } x \in [S \to S] \\ x. & \text{otherwise.} \end{cases}$

$p \in [S \to S] \subseteq S$.

$\underline{p(p)} = f(\underline{p(p)})$. $\therefore$ $p(p)$ is a fixed point of f.

③ Sol. ..... Use domain theory and categorical fixed point thm.

(1) $\overset{V_1}{\overset{\shortparallel}{D_1}} \simeq [D_1 \to_c D_1]$ (D_1 is a domain containing more than 1 element).

$V_1 \overset{\text{def.}}{=} D_1$ → slight generalisation of CFT.

(2) $\overset{(V_2)_\bot}{\overset{\shortparallel}{D_2}} \simeq [D_2 \to_c D_2]_\bot$ .... $D_2 = (V_2)_\bot$ ⇒ normal

$V_2 \overset{\text{def.}}{=} D_2 \setminus \{\bot\}$. $V_2 \simeq [\boxed{(V_2)_\bot} \to_c (V_2)_\bot]$

CFT, Dom^{EP}

(3) $D_3 \overset{\text{def.}}{=} (V_3)_\bot$, $V_3 \simeq [\boxed{V_3} \to_c (V_3)_\bot]$ ..... ⇒ eager.

Ex. Lambda calculus under ① $\to (\xrightarrow{*})$, ② $\Rightarrow_{normal}$, ③ $\Rightarrow_{eager}$.

$\to$ $\xrightarrow{*}$ ———— (D_1, V_1)

$\Rightarrow_{normal}$ ———— (D_2, V_2) … figure out 1–1 map here.

$\Rightarrow_{eager}$ ———— (D_3, V_3)

($[\![e]\!] = [\![e']\!]$ if $e \to e'$)

(1) why? D_1 vs D_2, D_3. ...... $\perp$.

In D_1, $\perp \in D_1$ is essentially the same as $[d \in D_1 \mapsto \perp]$.

smallest element.

In D_2, D_3, they are different.

Consider. $\lambda x. \left(\underbrace{(\lambda y.\, y\, y)\ (\lambda y.\, y\, y)} \right)$ ..... doesn't have β-normal form.

$d \mapsto \perp$

But it is in canonical form.

$\not\Downarrow \left((\lambda y.\, y\, y)\ (\lambda y.\, y\, y) \right)$

$\neq \perp$ under eager or normal.

(2) Why $D_2 \cdots \Rightarrow_{normal}$ and $D_3 \cdots \Rightarrow_{eager}$?

non-term. possibly $\Rightarrow_{normal}$

$$(\lambda x. (\lambda y. y)) \; \underbrace{((\lambda z. (z\ z)) \; (\lambda z. (z\ z)))}$$

terminating comp.

(value or canonical form).

go on forever under $\Rightarrow_{eager}$.

④ Denotational Semantics Worked out.

(1) Lambda-Calculus under $\to$ ($\xrightarrow{\times}$).

$$D_1 \underset{\phi}{\overset{\psi}{\rightleftarrows}} [D_1 \to_c D_1]$$

$$[\![-]\!] : \langle exp \rangle \to [Env \to_c D_1] \quad , \quad \eta \in Env = [\langle var \rangle \to D_1].$$

$$\underset{D_1}{[\![x]\!]\eta} = \eta(x)$$

$$\underset{D_1}{[\![\lambda x. e]\!]\eta} = \psi(\lambda d \in D_1. [\![e]\!][\eta \mid x : d])$$

$$\underset{D_1}{[\![e_1 e_2]\!]\eta} = \phi(\underbrace{[\![e_1]\!]\eta}_{D_1})(\underbrace{[\![e_2]\!]\eta}_{D_1})$$

(2) L.C. under $\Rightarrow_{normal}$

canonical forms.

$D_2 = (V_2)_\perp$ $\qquad V_2 \underset{\psi}{\overset{\phi}{\rightleftarrows}} [\underset{D_2}{(V_2)_\perp} \rightarrow_c \underset{D_2}{(V_2)_\perp}]$

all expression

$[\![-]\!]_n : \langle exp \rangle \rightarrow [Env \rightarrow_c D_2]$

$Env = [\langle var \rangle \rightarrow D_2]$

$[\![x]\!]_n \eta = \eta(x)$

$[\![\lambda x. e]\!]_n \eta = \psi(\lambda d \in \overset{(V_2)_\perp}{D_2}.\ [\![e]\!]_n [\eta | x : d])$

$[\![e_1 e_2]\!]_n \eta = \begin{cases} \perp & \text{if } [\![e_1]\!]_n \eta = \perp \\ \phi([\![e_1]\!]_n \eta)([\![e_2]\!]_n \eta) & \text{otherwise} \end{cases}$

(3) L.C under $\Rightarrow_{eager}$

canonical forms.

$D_3 = (V_3)_\perp$ $\qquad V_3 \underset{\psi}{\overset{d}{\rightleftarrows}} [V_3 \rightarrow_c (V_3)_\perp]$

all expressions

$[\![-]\!]_e : \langle exp \rangle \rightarrow [Env \rightarrow_c D_3]$

$Env = [\langle var \rangle \rightarrow V_3]$

$[\![x]\!]_e \eta = \eta(x)$

$[\![\lambda x. e]\!]_e \eta = \psi(\lambda v \in V_3.\ [\![e]\!]_e [\eta | x : v])$

$[\![e_1 e_2]\!]_e \eta = \begin{cases} \perp & \text{if } [\![e_1]\!]_e \eta = \perp \text{ or } [\![e_2]\!]_e \eta = \perp \\ \phi([\![e_1]\!]_e \eta)([\![e_2]\!]_e \eta) & \text{otherwise} \end{cases}$

(3) Justification.

$e_1 \rightarrow^* e_2 \Rightarrow [\![e_1]\!] = [\![e_2]\!]$

$e_1 \Rightarrow_{normal} e_2 \Rightarrow [\![e_1]\!]_n = [\![e_2]\!]_n$

$e_1 \Rightarrow_{eager} e_2 \Rightarrow [\![e_1]\!]_e = [\![e_2]\!]_e$