CS 520

Theory of Programming Language

05/12 – 05/26, 2021

$\boxed{\text{Lambda Calculus}}$

1. Reminder.

① $\langle exp \rangle ::= \langle var \rangle \mid \lambda \langle var \rangle . \langle exp \rangle \mid \langle exp \rangle \langle exp \rangle$.

② FV, substitution. , α-equivalence. $(\lambda x . x)(\lambda y . z) \equiv (\lambda z . z)(\lambda y' . z)$

③ Contraction - reduction.
$\rightarrow$ $\quad\quad$ $\rightarrow^*$

β-reduction

$$\frac{}{(\lambda x.\, e)\ e' \rightarrow e/x \rightarrow e'}$$

Renaming

$$\frac{e \rightarrow e' \qquad e' \equiv e''}{e \rightarrow e''}$$

✓

$\boxed{\begin{array}{l}\text{* Contextual closure.} \\ \dfrac{e_0' = \ldots\ e_0 \ldots \qquad e_0 \rightarrow e_1 \qquad \ldots e_1 \ldots = e_1'}{e_0' \rightarrow e_1'}\end{array}}$

④ β-normal form. e if $e \not\rightarrow e'$ for any e'.

(1) ⑤ Given e, is it possible to get two diff. β-normal-form expressions from e using $\rightarrow$?

No. ... Church-Rosser Thm.

(2) Given e, can we always get a β-normal form e' from e using $\rightarrow$?

No. ..... $(\lambda x.\, x\ x)\ (\lambda z.\, z\ z) =: e$.

$\downarrow$

$(\lambda z.\, z\ z)\ (\lambda z.\, z\ z)$.

(3). Given e, if $e \rightarrow^* e'$ for some β-normal form e', then can we find e' systematically?

Yes. ... normal-order reduction.

2. Church-Rosser and unique β-normal form.

[Church-Rosser] e ... expression in lamda cal.

$e \xrightarrow{*} e_1'$,

$e \xrightarrow{*} e_2'$ for some expressions e_1' and e_2'

$\Rightarrow \exists e_0$ s.t. $e_1' \xrightarrow{*} e_0$ and $e_2' \xrightarrow{*} e_0$

$\ulcorner$

$$
\begin{tikzcd}
 & e \arrow[dl, "*"'] \arrow[dr, "*"] & \\
e_1' \arrow[dr, dotted, "*"'] & & e_2' \arrow[dl, dotted, "*"] \\
 & e_0 &
\end{tikzcd}
$$

.... diagrammatic rewriting of the thm $\lrcorner$.

Prop 10.3. If $e \xrightarrow{*} e_1$ and $e \xrightarrow{*} e_2$ for β-normal form exprs. e_1 and e_2, then $e_1 \equiv e_2$ (i.e., e_1 and e_2 are α-equivalent).

Proof. By assm., $e \xrightarrow{*} e_1$ and $e \xrightarrow{*} e_2$. By CR., $\exists e_0$ s.t. $e_1 \xrightarrow{*} e_0$ and $e_2 \xrightarrow{*} e_0$

But e_1 and e_2 are in β-normal form. $e_1 \equiv e_0$ and $e_2 \equiv e_0$.

$\therefore$ $e_1 \equiv e_2$. $\square$

3. Normal-order reduction. ... a particular way of applying $\to$

$\left(\xrightarrow{}^{*}_{normal} \subseteq \xrightarrow{}^{*} \right)$ deterministic/algorithmic

$(\lambda u. \underline{\lambda v. v}) \left(\underline{(\lambda x. x\ x)\ (\lambda x. x\ x)} \right) \to (\lambda v. v)$

$\hookrightarrow (\lambda u. \lambda v. v) \left((\lambda x. x\ x)\ (\lambda x. x\ x) \right)$

$\hookrightarrow$

"

① Normal-order reduction ..... best in terms of getting β-normal-form answers.

② ... leftmost and outermost reduction. contraction.

$$\left(\overline{(\lambda x. x)\ (\lambda y. y)} \right) \left(\overline{(\lambda x. x)\ (\lambda y. y)} \right).$$

$\leadsto$ not β-redex

$$\to \underline{(\lambda y. y)} \left(\underline{(\lambda x. x)\ (\lambda y. y)} \right)$$

$$\to \underline{(\lambda x. x)\ (\lambda y. y)} \to (\lambda y. y)$$

$\xrightarrow[\text{normal}]{}$, $\xrightarrow[\text{normal}]{*}$

Thm. If $e \xrightarrow{*} e'$ for β-normal e', then $e \xrightarrow[\text{normal}]{*} e'$.

Note: $\xrightarrow[\text{normal}]{}$ is deterministic modulo $\equiv$.

K. Evaluation. ...... restricted version of $\to^*$ that corresponds to the run-time of a PL.

① which restrictions? (1). Only evaluate expressions w/o free vars.
closed expressions.

✓ (2) Don't evaluate the body of a fn unless the function is applied.

$$\lambda x.\ \underbrace{\underline{(\lambda y.\, y)\ x}}\ \longrightarrow\ \lambda x.\, x$$

② Evaluation relation. $\Rightarrow \subseteq \langle exp \rangle \times \langle exp \rangle$. ..... $\left[\xrightarrow{*} \right.$ independent def'n. ($\rightarrow$ equiv.

$\vdots$

(1) captures (1) & (2). from above.

(2). describes big-step computation.

③ Normal-order eval. (relation) $\Rightarrow_{normal} \subseteq \langle \text{closed exp} \rangle \times \langle \text{canonical-form exp} \rangle$.

$\langle \text{cform} \rangle = \lambda \langle var \rangle . \langle exp \rangle$. .... canonical form.

(value expressions).

$\xrightarrow{*}$

Canonical forms.

$$\frac{}{\lambda x.\, e \Rightarrow_{normal} \lambda x.\, e}$$

Application. (β-evaluation).

$$\frac{e \Rightarrow_{normal} \lambda x.\, e_0 \qquad e_0/x \to e' \Rightarrow_{normal} z}{e\, e' \Rightarrow_{normal} z}$$

$$\dfrac{\dfrac{}{(\lambda x.\,(\lambda y.y)\,x) \Rightarrow_{normal} (\lambda x.\,(\lambda y.y)^{*}\,x)} \qquad \dfrac{\dfrac{}{(\lambda y.y) \Rightarrow_{normal} (\lambda y.y)} \qquad \dfrac{}{(\lambda z.z) \Rightarrow_{normal} (\lambda z.z)}}{(\lambda y.y)\,(\lambda z.z) \Rightarrow_{normal} (\lambda z.z)}}{(\lambda x.\,(\lambda y.y)\,x)\,(\lambda z.z) \Rightarrow_{normal} (\lambda z.z)}$$

Any relationship between $\Rightarrow_{normal}$ and $\xrightarrow{*}$?

Yes, there is. $\xrightarrow{*}_{\text{normal, w/o contracting inside } \lambda}$ ... nr

Prop 10.6. For all closed e and canonical z

$$e \xrightarrow{*}_{nr} z \quad \text{iff} \quad e \Rightarrow_{normal} z.$$

Note: Normal-order ev. forms the basis of Haskell, Scala.

(1) Eager evaluation $\Rightarrow_{eager} \subseteq \langle \text{closed exp} \rangle \times \langle \text{cform} \rangle$.

1-contraction step

$$(\lambda x.\ x\ x)\ \big(\underbrace{(\lambda y.\ y)\ (\lambda z.\ z)}_{*}\big) \Rightarrow_{normal} \cdots\cdots$$

2 times.

1-contraction step.

Canonical Form

$$\frac{}{\lambda x.\ e \Rightarrow_{eager} \lambda x.\ e}$$

β_E-evaluation

$$\frac{e_0 \Rightarrow_{eager} \lambda x.\ e_0' \qquad e_1 \Rightarrow_{eager} z \qquad e_0'/x \to z \Rightarrow_{eager} z'}{e_0\ e_1 \Rightarrow_{eager} z'}$$

* Performance in terms of # of β-eval / β_E-eval steps ... no ~~clear~~ theoretical winner.

Understandability ... $\Rightarrow_{eager}$ easier.

Theoretical property ... $\Rightarrow_{normal}$ behaves better.

$(\rightarrow^*_{normal})$