Lambda Calculus

1. Comparing λ-terms

1.

$$=_\alpha \subset =_\beta \subset \cong \subset \Lambda^2$$

2.

x	x	$=_\alpha$
x	y	Λ^2
I	SKK	$=_\beta$
YI	Ω	$=_\beta$
$\lambda x.\lambda x.\lambda y.x$	$\lambda x.\lambda y.\lambda z.y$	$=_\alpha$
$\lambda x.\lambda x.\lambda y.x$	$\lambda x.\lambda y.\lambda z.x$	Λ^2
$Y(\lambda f.\lambda x.Kx(fx))y$	Ω	Λ^2
$\lambda x.fx$	f	$\cong$
Y	Ω	Λ^2

2. Normal forms

- $\mathcal{N}' \subset \mathcal{N}$:

  Let $M \in \mathcal{N}'$. We show that $M \in \mathcal{N}$ by rule induction on $M \in \mathcal{N}'$.

  - If $M \in \mathcal{N}'$ by (Var). Then clearly, $M \in \mathcal{N}$ since $\mathcal{V} \subset \mathcal{N}$.
  - If $M \in \mathcal{N}'$ by (Abs). Then, $M = \lambda x.M'$ for some $M' \in \mathcal{N}'$. By induction hypothesis, $M' \in \mathcal{N}$. So, $\lambda x.M' = M$ is also in $\mathcal{N}$.
  - If $M \in \mathcal{N}'$ by (App$_n$) for some $n \in \mathbb{N}$, $n > 0$. Then $M = xM_1 \cdots M_n$ for some $x \in \mathcal{V}$ and $M_i \in \mathcal{N}'$. By induction hypothesis, we have that $M_i \in \mathcal{N}$ for every i. Now, since $M = xM_1 \cdots M_n$, M is not itself a β-redex and by assumption, none of the M_i contains a β-redex, so M is in normal form: $M \in \mathcal{N}$.

- $\mathcal{N} \subset \mathcal{N}'$

  Let $M \in \mathcal{N}$. We show that $M \in \mathcal{N}'$ by induction on M.

  - If $M = x \in \mathcal{V}$, then by (Var), $M \in \mathcal{N}'$.
  - If $M = \lambda x.M'$. Since $M \in \mathcal{N}$, we also have that $M' \in \mathcal{N}$ (otherwise, M contains a β-redex). By induction hypothesis, we then have that $M' \in \mathcal{N}'$. By rule (Abs), we then have that $M \in \mathcal{N}'$.
  - If $M = M_1M_2$. Since, $M \in \mathcal{N}$, we also have that $M_1 \in \mathcal{N}$ and $M_2 \in \mathcal{N}$. By induction hypothesis, we have then that $M_1 \in \mathcal{N}'$ and $M_2 \in \mathcal{N}'$. Now, since $M = M_1M_2$ is a normal form, M_1 cannot be an abstraction (otherwise M is β-reducible). So $M_1 \in \mathcal{N}'$ by (Var) or (App$_n$) for some positive integer n.
    - If $M_1 \in \mathcal{N}'$ by (Var). Then (App$_1$) allows us to conclude that $M = M_1M_2 \in \mathcal{N}'$.
    - If $M_1 \in \mathcal{N}'$ by (App$_n$) for some n. Then $M_1 = xM'_1 \cdots M'_n$ for some $M'_i \in \mathcal{N}'$. The rule (App$_{n+1}$) allows us to conclude that $M = M_1M_2 = xM'_1 \cdots M'_nM_2 \in \mathcal{N}'$.

Pi Calculus

3. Transition and transition graph

1.

To shorten the derivation tree we define two processes P and Q:

$$\begin{array}{rcl} P & = & c(c).\overline{c}\langle z\rangle.\,\mathbf{0} + (\nu z)\,\overline{c}\langle z\rangle.z(z).\overline{z}.\,\mathbf{0} \\ Q & = & (\nu x)\,(\overline{x}\langle z\rangle.\,\mathbf{0} \mid x(z).\overline{z}.\,\mathbf{0}) \mid !\,P \end{array}$$

$$\text{(Res)}\,\dfrac{\text{(Rep)}\,\dfrac{\text{(Close)}\,\dfrac{\text{(Sum)}\,\dfrac{\text{(Inp)}\,\dfrac{}{c(c).\overline{c}\langle z\rangle.\,\mathbf{0} \xrightarrow{c\,x} \overline{x}\langle z\rangle.\,\mathbf{0}}}{P \xrightarrow{c\,x} \overline{x}\langle z\rangle.\,\mathbf{0}} \qquad \text{(Rep)}\,\dfrac{\text{(Par)}\,\dfrac{\text{(Sum)}\,\dfrac{\text{(Alp)}\,\dfrac{\text{(Open)}\,\dfrac{\text{(Out)}\,\dfrac{}{\overline{c}\langle x\rangle.x(z).\overline{z}.\,\mathbf{0} \xrightarrow{\overline{c}\,x} x(z).\overline{z}.\,\mathbf{0}}}{(\nu x)\,\overline{c}\langle x\rangle.x(z).\overline{z}.\,\mathbf{0} \xrightarrow{(\nu x)\overline{c}\,x} x(z).\overline{z}.\,\mathbf{0}}}{(\nu z)\,\overline{c}\langle z\rangle.z(z).\overline{z}.\,\mathbf{0} \xrightarrow{(\nu x)\overline{c}\,x} x(z).\overline{z}.\,\mathbf{0}}}{P \xrightarrow{(\nu x)\overline{c}\,x} x(z).\overline{z}.\,\mathbf{0}}}{P \mid !\,P \xrightarrow{(\nu x)\overline{c}\,x} x(z).\overline{z}.\,\mathbf{0} \mid !\,P}}{!\,P \xrightarrow{(\nu x)\overline{c}\,x} x(z).\overline{z}.\,\mathbf{0} \mid !\,P}}{P \mid !\,P \xrightarrow{\tau} (\nu x)\,(\overline{x}\langle z\rangle.\,\mathbf{0} \mid x(z).\overline{z}.\,\mathbf{0} \mid !\,P)}}{!\,P \xrightarrow{\tau} (\nu x)\,(\overline{x}\langle z\rangle.\,\mathbf{0} \mid x(z).\overline{z}.\,\mathbf{0} \mid !\,P)}}{(\nu c)\,!\,P \xrightarrow{\tau} (\nu c)\,(\nu x)\,(\overline{x}\langle z\rangle.\,\mathbf{0} \mid x(z).\overline{z}.\,\mathbf{0} \mid !\,P)}$$

or (if structural congruence is allowed):

$$\text{(Res)}\,\dfrac{\text{(Rep)}\,\dfrac{\text{(Rep)}\,\dfrac{(\equiv)\,\dfrac{\text{(Par)}\,\dfrac{\text{(Close)}\,\dfrac{\text{(Sum)}\,\dfrac{\text{(Inp)}\,\dfrac{}{c(c).\overline{c}\langle z\rangle.\,\mathbf{0} \xrightarrow{c\,x} \overline{x}\langle z\rangle.\,\mathbf{0}}}{P \xrightarrow{c\,x} \overline{x}\langle z\rangle.\,\mathbf{0}} \qquad \text{(Sum)}\,\dfrac{(\equiv)\,\dfrac{\text{(Open)}\,\dfrac{\text{(Out)}\,\dfrac{}{\overline{c}\langle x\rangle.x(z).\overline{z}.\,\mathbf{0} \xrightarrow{\overline{c}\,x} x(z).\overline{z}.\,\mathbf{0}}}{(\nu x)\,\overline{c}\langle x\rangle.x(z).\overline{z}.\,\mathbf{0} \xrightarrow{(\nu x)\overline{c}\,x} x(z).\overline{z}.\,\mathbf{0}}}{(\nu z)\,\overline{c}\langle z\rangle.z(z).\overline{z}.\,\mathbf{0} \xrightarrow{(\nu x)\overline{c}\,x} x(z).\overline{z}.\,\mathbf{0}}}{P \xrightarrow{(\nu x)\overline{c}\,x} x(z).\overline{z}.\,\mathbf{0}}}{P \mid P \xrightarrow{\tau} (\nu x)\,(\overline{x}\langle z\rangle.\,\mathbf{0} \mid x(z).\overline{z}.\,\mathbf{0})}}{(P \mid P) \mid !\,P \xrightarrow{\tau} Q}}{P \mid (P \mid !\,P) \xrightarrow{\tau} Q}}{P \mid !\,P \xrightarrow{\tau} Q}}{!\,P \xrightarrow{\tau} Q}}{(\nu c)\,!\,P \xrightarrow{\tau} (\nu c)\,Q}$$

2.

$$\begin{tikzcd} \mathrm{W}(l) \arrow[dr, bend left, "(\nu t{,}f)\,\overline{l}\,t{,}f"] & & \mathbf{0} \\ & t.\mathrm{W}(l) + f.\,\mathbf{0} \arrow[ul, bend left, "t"] \arrow[ur, bend right, "f"'] & \end{tikzcd}$$

4. Church encoding

$$\mathrm{D}(l,k) \quad := \quad l(z, sd, d).\overline{d}\langle k\rangle.\,\mathbf{0}$$

To ease the writing of processes, we define three well known integers:

$$\begin{array}{rcl} \mathrm{Zero}(l) & := & \mathrm{Z}(l) \\ \mathrm{One}(l) & := & (\nu k)\,(\mathrm{SD}(l,k) \mid \mathrm{Zero}(k)) \\ \mathrm{Two}(l) & := & (\nu k)\,(\mathrm{D}(l,k) \mid \mathrm{One}(k)) \end{array}$$

$$\mathrm{Copy}(l,k) \quad := \quad (\nu caseZ, caseSD, caseD) \quad \begin{array}{l} (\overline{l}\langle caseZ, caseSD, caseD\rangle.\,\mathbf{0} \mid \\ \quad caseZ.(\mathrm{Zero}(l) \mid \mathrm{Zero}(k)) \\ + caseSD(l').(\mathrm{SD}(l,l') \mid (\nu k')\,(\mathrm{SD}(k,k') \mid \mathrm{Copy}(l',k'))) \\ + caseD(l').(\mathrm{D}(l,l') \mid (\nu k')\,(\mathrm{D}(k,k') \mid \mathrm{Copy}(l',k'))) \end{array}$$

$$\mathrm{Succ}(l,k) \quad := \quad (\nu caseZ, caseSD, caseD) \quad \begin{array}{l} (\overline{l}\langle caseZ, caseSD, caseD\rangle.\,\mathbf{0} \mid \\ \quad caseZ.\mathrm{One}(k) \\ + caseSD(l').(\nu k')\,(\mathrm{D}(k,k') \mid \mathrm{Succ}(l',k')) \\ + caseD(l').\mathrm{SD}(k,l')) \end{array}$$