輪講：Type Systems and Programming Languages

Mizukami Tatsuo

平成 13 年 5 月 7 日

1　普通の λ 式を名前なし λ 式に変換する方法

まず最初に Context Γ についての演算子の定義を定義する。Context $\Gamma = x_{n-1}, x_{n-2}, \ldots, x_0$ とすると、演算子 index, push を

- $\mathrm{index}_\Gamma(i) = x_i$
- $\mathrm{push}_\Gamma(u) = \text{Context } \Gamma' : x_{n-1}, \ldots, x_0, u$

と定義する。

Ordinary term はつぎのようにして定義されるので

$$
\begin{array}{rll}
\text{terms } t := & & \\
 & x & \text{variable} \\
 & \lambda x.t & \text{abstraction} \\
 & t\ t & \text{application}
\end{array}
$$

Ordinary term から Nameless term への変換関数 removenames は次のようにして定義される。

$$
\begin{array}{rcl}
\mathrm{removenames}_\Gamma(x) & = & i \quad \text{where } \mathrm{index}_\Gamma(i) = x,\ \forall j < i,\ \mathrm{index}_\Gamma(j) \neq x \\
\mathrm{removenames}_\Gamma(\lambda x.t) & = & \lambda.\ \mathrm{removenames}_{\Gamma'}(t) \quad \text{where } \Gamma' = \mathrm{push}_\Gamma(x) \\
\mathrm{removenames}_\Gamma(t_1\ t_2) & = & \mathrm{removenames}_\Gamma(t_1)\ \mathrm{removenames}_\Gamma(t_2)
\end{array}
$$

Nameless term も Ordinary term と同じように定義されるであろうから、逆に Nameless term から Ordinary term への変換関数 restorenames は次のようにして定義される。

$$
\begin{aligned}
\mathrm{restorenames}_\Gamma(k) &= \mathrm{index}_\Gamma(k) \\
\mathrm{restorenames}_\Gamma(\lambda.t) &= \lambda u.\ \mathrm{restorenames}_{\Gamma'}(t) \\
&\qquad \text{where } \Gamma' = \mathrm{push}_\Gamma(u),\ u \in V,\ u \notin \mathrm{dom}(\Gamma) \\
\mathrm{restorenames}_\Gamma(t_1\ t_2) &= \mathrm{restorenames}_\Gamma(t_1)\ \mathrm{restorenames}_\Gamma(t_2)
\end{aligned}
$$

V は変数の集合であり、「 $u \in V,\ u \notin \mathrm{dom}(\Gamma)$ 」という操作は新たな変数 u をとってくるということを表している。上記の定義から orinary $\to$ nameless $\to$ ordinary, nameless $\to$ orinary $\to$ nameless の操作である

$$
\begin{aligned}
\mathrm{removenames}_\Gamma(\mathrm{restorenames}_\Gamma(t)) &= t \\
\mathrm{restorenames}_\Gamma(\mathrm{removenames}_\Gamma(t)) &= t
\end{aligned}
$$

が成立するはずである。