练习**1.6**

$$(\lambda xy.x)(\lambda x.xx)(\lambda x.xx)$$

$$\to_\beta (\lambda y.(\lambda x.xx))(\lambda x.xx)$$

$$\to_\beta \lambda x.xx$$

因此可以归约到范式

$$(1)\quad (\lambda xy.x)(\lambda x.xx)(\lambda x.xx)$$

$$(\lambda xy.x)((\lambda x.xx)(\lambda x.xx))$$

$$\to_\beta (\lambda x.xx)(\lambda x.xx)$$

$$\to_\beta (\lambda x.xx)(\lambda xx.xx)$$

...

因此不能归约到任何范式

$$(2)\quad (\lambda xy.x)((\lambda x.xx)(\lambda x.xx))$$

$$(\lambda x.x)(\lambda y.yyy)(\lambda xy.x)$$
$$\to_\beta (\lambda y.yyy)(\lambda xy.x)$$
$$\to_\beta (\lambda xy.x)(\lambda xy.x)(\lambda xy.x)$$
$$\to_\beta (\lambda xy.x)(\lambda xy.x)$$
$$\to_\beta \lambda xy.x$$

因此可以归约到范式

(3) $(\lambda x.x)(\lambda y.yyy)(\lambda xy.x)$

$$(\lambda fx.fxx)(\lambda xy.y)y$$
$$\to_\beta (\lambda x.(\lambda xy.y)xx)y$$
$$\to_\beta (\lambda xy.y)yy$$
$$\to_\beta (\lambda y.y)y$$
$$\to_\beta y$$

因此可以归约到范式

(4) $(\lambda fx.fxx)(\lambda xy.y)y$

练习1.18

练习 1.18. 斯科特数（*Scott numerals*）是对自然数的另一种编码。

- 数字零 $\mathbf{Z} := \lambda z.\lambda s.\, z$
- 后继函数 $\mathbf{S} := \lambda n.\lambda z.\lambda s.\, s\, n$
- 模式匹配 $\mathbf{caseN} := \lambda n.\lambda u.\lambda f.\, n\, u\, f$

斯科特编码的自然数有下列形式的范式：

$$\begin{aligned} \mathbf{one} &:= \lambda z.\lambda s.\, s\,\mathbf{Z} \\ \mathbf{two} &:= \lambda z.\lambda s.\, s\,\mathbf{one} \\ \mathbf{three} &:= \lambda z.\lambda s.\, s\,\mathbf{two} \\ &\vdots \end{aligned}$$

为得到斯科特编码的加法，我们注意到

$$\mathbf{add}\, n\, m = \mathbf{caseN}\, n\, m\, (\lambda n'.\,\mathbf{S}\,(\mathbf{add}\, n'\, m))$$

验证这个等式成立，并借助一个不动点写出用 λ-项表示的加法函数。

一、$n=Z$ 时

$$\begin{aligned} \mathrm{add}\, n\, m &= \mathrm{caseN}\, n\, m\, (\lambda n'.S(\mathrm{add}\, n'\, m)) \\ &= Z\, m\, (\lambda n'.S(\mathrm{add}\, n'\, m)) \\ &= (\lambda z.\lambda s.z)\, m\, (\lambda n'.S(\mathrm{add}\, n'\, m)) \\ &\to_\beta (\lambda s.m)(\lambda n'.S(\mathrm{add}\, n'\, m)) \\ &\to_\beta m \end{aligned}$$

练习 1.18. 斯科特数（*Scott numerals*）是对自然数的另一种编码。

- 数字零 $\mathbf{Z} := \lambda z.\lambda s.\, z$
- 后继函数 $\mathbf{S} := \lambda n.\lambda z.\lambda s.\, s\, n$
- 模式匹配 $\mathbf{caseN} := \lambda n.\lambda u.\lambda f.\, n\, u\, f$

斯科特编码的自然数有下列形式的范式：

$$\begin{array}{rcl}\mathbf{one} & := & \lambda z.\lambda s.\, s\, \mathbf{Z} \\ \mathbf{two} & := & \lambda z.\lambda s.\, s\, \mathbf{one} \\ \mathbf{three} & := & \lambda z.\lambda s.\, s\, \mathbf{two} \\ & \vdots & \end{array}$$

为得到斯科特编码的加法，我们注意到

$$\mathbf{add}\, n\, m = \mathbf{caseN}\, n\, m\, (\lambda n'.\, \mathbf{S}\, (\mathbf{add}\, n'\, m))$$

验证这个等式成立，并借助一个不动点写出用 λ-项表示的加法函数。

二. 当 $n \neq Z$ 时，由 Scott numerals 的递归定义，一定存在另一个 Scott numeral 使得 $n = Sn'$，这里的 n' 对应自然数 $n-1$

此时 $addnm = caseN\, n m\, (\lambda n'. S(add\, n' m))$

$\to_\beta (Sn')\, m\, (\lambda n'. S(add\, n' m))$

$= ((\lambda n.\lambda z.\lambda s. s n) n')\, m\, (\lambda n'. S(add\, n' m))$

$\to_\beta (\lambda z.\lambda s. s n')\, m\, (\lambda n'. S(add\, n' m))$

$\to_\beta (\lambda s. s n')(\lambda n'. S(add\, n' m))$

$\to_\beta (\lambda n'. S(add\, n' m))\, n'$

$\to_\beta S\,(add\, n' m)$

练习 1.18. 斯科特数（*Scott numerals*）是对自然数的另一种编码。

- 数字零 $\mathbf{Z} := \lambda z.\lambda s.\, z$
- 后继函数 $\mathbf{S} := \lambda n.\lambda z.\lambda s.\, s\, n$
- 模式匹配 $\mathbf{caseN} := \lambda n.\lambda u.\lambda f.\, n\, u\, f$

斯科特编码的自然数有下列形式的范式：

$$\begin{array}{rcl}\mathbf{one} & := & \lambda z.\lambda s.\, s\, \mathbf{Z} \\ \mathbf{two} & := & \lambda z.\lambda s.\, s\, \mathbf{one} \\ \mathbf{three} & := & \lambda z.\lambda s.\, s\, \mathbf{two} \\ & \vdots & \end{array}$$

为得到斯科特编码的加法，我们注意到

$$\mathbf{add}\, n\, m = \mathbf{caseN}\, n\, m\, (\lambda n'.\, \mathbf{S}\, (\mathbf{add}\, n'\, m))$$

验证这个等式成立，并借助一个不动点写出用 λ-项表示的加法函数。

如果将 addnm 看作一个函数 $add(n,m)$

则 $add(n,m) = \begin{cases} m, & n=0 \\ add(n-1, m), & n \geq 1 \end{cases}$

解得 $add(n,m) = m+n$

练习 1.18. 斯科特数（*Scott numerals*）是对自然数的另一种编码。

- 数字零 $\mathbf{Z} := \lambda z.\lambda s.\, z$
- 后继函数 $\mathbf{S} := \lambda n.\lambda z.\lambda s.\, s\, n$
- 模式匹配 $\mathbf{caseN} := \lambda n.\lambda u.\lambda f.\, n\, u\, f$

斯科特编码的自然数有下列形式的范式：

$$\begin{array}{rcl}\mathbf{one} & := & \lambda z.\lambda s.\, s\, \mathbf{Z} \\ \mathbf{two} & := & \lambda z.\lambda s.\, s\, \mathbf{one} \\ \mathbf{three} & := & \lambda z.\lambda s.\, s\, \mathbf{two} \\ & \vdots & \end{array}$$

为得到斯科特编码的加法，我们注意到

$$\mathbf{add}\, n\, m = \mathbf{caseN}\, n\, m\, (\lambda n'.\, \mathbf{S}\, (\mathbf{add}\, n'\, m))$$

验证这个等式成立，并借助一个不动点写出用 λ-项表示的加法函数。

二、用不动点写出用λ-项表示的加法函数

$addnm = caseN\, nm\, (\lambda n'. S(add\, n' m))$

$add = \lambda n.\lambda m. caseN\, nm\, (\lambda n'. S(add\, n' m))$

$add = (\lambda f.\lambda n.\lambda m. caseN\, nm\, (\lambda n'. S(add\, n' m)))\, add$

令 $F \overset{def}{=} \lambda f.\lambda n.\lambda m.\, caseN\, nm\, (\lambda n'. S(add\, n' m))$

则 $add = F\, add$

则 $add \overset{def}{=} \Theta F$

$= \Theta(\lambda f.\lambda n.\lambda m.\, caseN\, nm\, (\lambda n'. S(add\, n' m)))$