

# EXERCISES

## CHAPTER 1

SEAN LI <sup>1</sup>

### 1. Reduced

#### Problem

(1.1) Simplify notation of the following terms

(a)  $(\lambda x.((xz)y)(xx))$

(b)  $((\lambda x.(\lambda y.(\lambda z.(z((xy)z)))))(\lambda u.u))$

*Solution.*

(a)  $\lambda x.(xzy)(xx)$

(b)  $(\lambda xyz.x(xyz))(\lambda u.u)$

#### Problem

(1.2) Find the alpha equivalent terms to

$\lambda x.x(\lambda x.x)$

In

(a)  $\lambda y.y(\lambda x.x)$

(b)  $\lambda y.y(\lambda x.y)$

(c)  $\lambda y.y(\lambda x.y)$

*Solution.* Only (a).

### Problem

(1.3) Prove

$$\lambda x.x(\lambda z.y) \underset{\alpha}{=} \lambda z.z(\lambda z.y)$$

*Solution.*

*Proof.* By definition of alpha equivalence

$$M \underset{\alpha}{=} N \iff \exists \varphi, M^\varphi \underset{\alpha}{\rightarrow} N \wedge \text{FR } M = \text{FR } N$$

The witness of  $\varphi$  is substituting bound variable  $x$  with  $z$ , and  $z$  is not a free variable in the term, thus the two terms are alpha equivalent.

$$\lambda x.x(\lambda z.y) \xrightarrow[\alpha]{x \rightarrow z} \lambda z.z(\lambda z.y)$$

■

### Problem

(1.4) Consider the following term:

$$U := (\lambda z.zxz)((\lambda y.xy)x)$$

1. Find Sub  $U$
2. Draw tree rep of  $U$
3. Find FV  $U$
4. Find alpha equivalent terms to  $U$  from below and point out which of those follows the Barendregt convention:

(a)  $(\lambda y.yxy)((\lambda z.xz)x)$

(b)  $(\lambda x.xyx)((\lambda z.yz)y)$

(c)  $(\lambda y.yxy)((\lambda y.xy)x)$

(d)  $(\lambda v.(vx)v)((\lambda u.uv)x)$

1. Find Sub  $U$ .

*Solution.*

Sub  $U =$

$$\begin{aligned} & \{(\lambda z.zxz)((\lambda y.xy)x), (\lambda z.zxz), ((\lambda y.xy)x)\} \cup \\ & \{(\lambda y.xy), (y), (\lambda z.xz), (x)\} \cup \\ & \{(\lambda y.x), (y)\} \cup \{(\lambda z.x), (z)\} \cup \{(y), (x)\} \\ & = \{(\lambda z.zxz)((\lambda y.xy)x), (\lambda z.zxz), (\lambda y.xy)x, \\ & (\lambda y.xy), (\lambda z.xz), (\lambda y.x), (\lambda z.x), y, x\} \end{aligned}$$

2. Draw a tree rep of  $U$ .

*Solution.*



3. Find FV  $U$

*Solution.*

$$\begin{aligned} \text{FV } U &= \text{FV } (\lambda y.yxy) \cup \text{FV } (\lambda z.xz)x \\ &= (\text{FV } yxy) \setminus \{y\} \cup (\text{FV } \lambda z.xz) \cup \{x\} \\ &= (\text{FV } yx) \setminus \{y\} \cup (\text{FV } xz) \setminus \{z\} \cup \{x\} \\ &= \{x\} \end{aligned}$$

4. Find an alpha-equivalent term.

*Solution.*

$$(a) \stackrel{=}{\alpha} (c) \stackrel{=}{\alpha} U$$

Only (a) follows the Barendregt convention.

### Problem

(1.5) Give the results of the following substitutions

- (a)  $(\lambda x.y(\lambda y.xy))[y := \lambda z.zx]$
- (b)  $((xyz)[x := y])[y := z]$
- (c)  $((\lambda x.xyz)[x := y])[y := z]$
- (d)  $(\lambda y.yyx)[x := yz]$

*Solution.*

- (a)  $(\lambda v.(\lambda z.zx)(\lambda u.vu))$
- (b)  $(yyz)[y := z] = zzz$
- (c)  $(\lambda x.xyz)[y := z] = (\lambda x.xzz)$
- (d)  $(\lambda u.uu(yz))$

### Problem

(1.6)

$$\neg \left( \forall M L N \in \Lambda, M[x := N, y := L] \equiv_{\alpha} M[x := N][y := L] \right)$$

*Solution.*

*Proof.* Because  $\text{RHS} = M[x := N][y := L] = M[x := N[y := L]][y := L]$ , if  $y \in \text{FV } N$ , then what  $x$  gets substituted with will have  $y$  substituted for  $L$ , which is completely different with LHS. ■

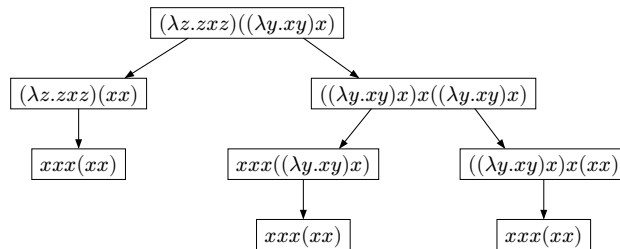
### Problem

(1.7) Find all available redexes in

$$U := (\lambda z.zxz)((\lambda y.xy)x)$$

And all reduction pathes to the  $\beta$ -normal form.

*Solution.* The first redex is the term as an application itself; another the second term in the application.



### Problem

(1.8) Show that

$$(\lambda x.xx)y \not\equiv_{\beta} (\lambda xy.yx)xx$$

*Solution.* By Corollary 1.9.9, it suffices to prove the hypothesis with a proof of a common normal reduced form from LHS and RHS not existing.

*Contradiction.* By definition of  $\equiv_{\beta}$ , there exists The set of all terms attainable from  $\beta$ -reduction on  $(\lambda x.xx)y$  and  $(\lambda xy.yx)xx$  do not intersect. Therefore,

$$\neg \left( \exists L \in \Lambda, (\lambda x.xx)y \rightarrow_{\beta}^* L \wedge (\lambda xy.yx)xx \rightarrow_{\beta}^* L \right) \implies \neg \left( (\lambda x.xx)y \equiv_{\beta} (\lambda xy.yx)xx \right)$$

■

### Problem

(1.9) Define the combinators

$$K := \lambda xy.x$$

$$S := \lambda xyz.xz(yz)$$

Prove that

$$\forall PQ \in \Lambda, K PQ \rightarrow_{\beta}^* P$$

$$\forall PQR \in \Lambda, SPQR \rightarrow_{\beta}^* PR(QR)$$

*Solution.*

*Proof.*

$$K PQ = (\lambda xy.x)PQ \rightarrow_{\beta} (\lambda y.x)[x := P]Q \rightarrow_{\beta} P[y := Q] = P$$

$$SPQR = (\lambda xyz.xz(yz)) \rightarrow_{\beta}^* (xz(yz))[x := P][y := Q][z := R] = PR(QR)$$

■

### Problem

(1.10) We define the church numerals

$$\begin{aligned}\text{zero} &:= \lambda f x. x \\ \text{one} &:= \lambda f x. f x \\ \text{two} &:= \lambda f x. f f x \\ &\dots \\ \text{num}_n &:= \lambda f x. f^n x\end{aligned}$$

And operations

$$\begin{aligned}\text{add} &:= \lambda n m f x. m f (n f x) \\ \text{mul} &:= \lambda n m f x. m (n f) x\end{aligned}$$

Show

- (a)  $\text{add one one} \xrightarrow[\beta]{} \text{two}$
- (b)  $\text{add one one} \not\xrightarrow[\beta]{} \text{mul one zero}$

*Solution.*

$$\begin{aligned}\text{(a) } \text{add one one} &= (\lambda n m f x. m f (n f x)) (\lambda f x. f x) (\lambda f x. f x) \\ &\xrightarrow[\beta]{} (\lambda f x. (\lambda f x. f x) f ((\lambda f x. f x) f x)) \\ &\xrightarrow[\beta]{} (\lambda f x. (\lambda x. f x) f x) \\ &\xrightarrow[\beta]{} (\lambda f x. f f x) = \text{two} \\ \text{(b) } \text{mul one one} &= (\lambda n m f x. m (n f) x) (\lambda f x. f x) (\lambda f x. f x) \\ &\xrightarrow[\beta]{} \lambda f x. (\lambda f x. f x) ((\lambda f x. f x) f) x \\ &\xrightarrow[\beta]{} \lambda f x. f x = \text{one}\end{aligned}$$

Because no intermediate form in the beta reduction process of the two terms are  $\alpha$ -equivalent, by corollary 1.9.9 the two terms are not  $\beta$ -equivalent.

### Problem

(1.11) We define

$$\text{succ} := \lambda m f x. f(m f x) \text{ s.t. } \forall \text{num}_n, \text{succ num}_n = \text{num}_{n+1}$$

Prove

$$\text{succ zero} \stackrel{\beta}{=} \text{one}$$

$$\text{succ one} \stackrel{\beta}{=} \text{two}$$

*Solution.* It suffices to provide a witness of a reduction chain from one side to the other to prove  $\beta$ -equivalence.

*Proof.*

$$\begin{aligned} \text{succ zero} &= (\lambda m f x. f(m f x))(\lambda f x. x) \\ &\rightarrow_{\beta} (\lambda f x. f((\lambda f x. x) f x)) \\ &\rightarrow_{\beta} (\lambda f x. f x) = \text{one} \end{aligned}$$

The path  $\text{succ zero} \rightarrow_{\beta} \text{one}$  derived above is the witness of a reduction chain from LHS to RHS.

$$\begin{aligned} \text{succ one} &= (\lambda m f x. f(m f x))(\lambda f x. f x) \\ &\rightarrow_{\beta} (\lambda f x. f((\lambda f x. f x) f x)) \\ &\rightarrow_{\beta} (\lambda f x. f(f x)) = \text{two} \end{aligned}$$

The path  $\text{succ one} \rightarrow_{\beta} \text{two}$  derived above is the witness of a reduction chain from LHS to RHS. ■

### Problem

(1.12) We define the  $\lambda$ -terms  $\top_{\lambda}$  (true) and  $\perp_{\lambda}$  (false) and  $\neg_{\lambda}$  (not) by:

$$\begin{aligned} \top_{\lambda} &:= \lambda x y. x & \perp_{\lambda} &:= \lambda x y. y \\ \neg_{\lambda} &:= \lambda a. a \perp_{\lambda} \top_{\lambda} \end{aligned}$$

Show that

$$\neg_{\lambda}(\neg_{\lambda} \top_{\lambda}) \stackrel{\beta}{=} \top_{\lambda}$$

$$\neg_{\lambda}(\neg_{\lambda} \perp_{\lambda}) \stackrel{\beta}{=} \perp_{\lambda}$$

*Solution.* It suffices to provide a witness of a reduction chain from one side to the other to prove  $\beta$ -equivalence.

*Proof.*

$$\begin{aligned}
\neg_\lambda(\neg_\lambda \top_\lambda) &= \neg_\lambda((\lambda a. a \perp_\lambda \top_\lambda)(\lambda xy. x)) \\
&\xrightarrow[\beta]{\rightarrow} \neg_\lambda((\lambda xy. x) \perp_\lambda \top_\lambda) \\
&\xrightarrow[\beta]{\rightarrow} (\lambda a. a \perp_\lambda \top_\lambda) \perp_\lambda \\
&\xrightarrow[\beta]{\rightarrow} (\lambda xy. y) \perp_\lambda \top_\lambda \\
&\xrightarrow[\beta]{\rightarrow} \top_\lambda
\end{aligned}$$

$$\begin{aligned}
\neg_\lambda(\neg_\lambda \perp_\lambda) &= \neg_\lambda((\lambda a. a \perp_\lambda \top_\lambda)(\lambda xy. y)) \\
&\xrightarrow[\beta]{\rightarrow} \neg_\lambda((\lambda xy. x) \perp_\lambda \top_\lambda) \\
&\xrightarrow[\beta]{\rightarrow} (\lambda a. a \perp_\lambda \top_\lambda) \top_\lambda \\
&\xrightarrow[\beta]{\rightarrow} (\lambda xy. x) \perp_\lambda \top_\lambda \\
&\xrightarrow[\beta]{\rightarrow} \perp_\lambda
\end{aligned}$$

■

### Problem

(1.13) Define

$$\text{iszero} := \lambda m. m(\lambda x. \perp_\lambda) \top_\lambda$$

Prove

$$\text{iszero zero} \xrightarrow[\beta]{\rightarrow} \top_\lambda$$

$$\forall n \in \mathbb{N}^+, \text{iszero num}_n \xrightarrow[\beta]{\rightarrow} \perp_\lambda$$

*Solution.*

$$\begin{aligned}
\text{iszero zero} &= (\lambda m. m(\lambda x. \perp_\lambda) \top_\lambda)(\lambda fx. x) \\
&\xrightarrow[\beta]{\rightarrow} (\lambda fx. x)(\lambda x. \perp_\lambda) \top_\lambda \\
&\xrightarrow[\beta]{\rightarrow} \top_\lambda
\end{aligned}$$



$$\begin{aligned}
\text{iszero num}_n &= (\lambda m.m(\lambda x.\perp_\lambda)\top_\lambda)(\lambda f x.f^n x) \\
&\xrightarrow[\beta]{\rightarrow} (\lambda f x.f^n x)(\lambda x.\perp_\lambda)\top_\lambda \\
&\xrightarrow[\beta]{\rightarrow} (\lambda x.\perp_\lambda)((\lambda x.\perp_\lambda)^{n-1}\top_\lambda) \xrightarrow[\beta]{\rightarrow} \perp_\lambda
\end{aligned}$$

### Problem

(1.14) If-else can be modeled as

$$\text{ifelse} = \lambda x t f. x t f$$

Where when  $x$ , then  $t$ , else  $f$ . Prove correctness by applying  $\top_\lambda$  and  $\perp_\lambda$  on ifelse.

*Solution.*

$$\begin{aligned}
\text{ifelse } \top_\lambda &= (\lambda x t f. x t f)\top_\lambda \\
&\xrightarrow[\beta]{\rightarrow} (\lambda t f. (\lambda x y. x) t f) \xrightarrow[\beta]{\rightarrow} (\lambda t f. t) \\
\text{ifelse } \perp_\lambda &= (\lambda x t f. x t f)\perp_\lambda \\
&\xrightarrow[\beta]{\rightarrow} (\lambda t f. (\lambda x y. y) t f) \xrightarrow[\beta]{\rightarrow} (\lambda t f. f)
\end{aligned}$$

By applying the results to any two values, the correct corresponding value returns, ex, for ifelse  $\top_\lambda$ ,  $t$  is always returned.

### Problem

(1.15) Prove that  $\Omega := (\lambda x. x x)(\lambda x. x x)$  does not have a  $\beta$ -nf.

*Solution.* Firstly let's prove  $\Omega$ .

*Proof.* Induction on  $\Omega$ 's only reduction path proves that every  $\Omega \xrightarrow[\beta]{\rightarrow} \Omega_i = \Omega$ . For the base case because  $\Omega$  has one and only one redux, it could only reduce to  $\Omega_1$  which is equivalent to itself. For the inductive step,  $\Omega_i = \Omega$ , therefore  $\Omega_i \xrightarrow[\beta]{\rightarrow} \Omega_{i+1}$  is still  $\Omega$ .

By definition, a term having a  $\beta$ -nf requires the existence of a form in  $\beta$ -nf such that the term can reduce to. By induction,  $\Omega$  only reduces to  $\Omega$ , and  $\Omega$  is not in  $\beta$ -nf because it contains  $\beta$ -redex. Therefore,  $\Omega$  can never reduce to a  $\beta$ -nf, thus it does not have a  $\beta$ -nf. ■

### Problem

(1.16) Let  $M$  be a  $\lambda$ -term with the following properties:

- $M$  has a  $\beta$ -nf.
- There exists an infinite reduction path  $M \equiv M_0 \xrightarrow{\beta} M_1 \xrightarrow{\beta} \dots$  on  $M$ .

Prove that every  $M_i$  has a  $\beta$ -nf, and give an example of  $M$ .

*Solution.* An example would be  $(\lambda xy.y)\Omega$ . Reduction can go on infinitely by reducing on  $\Omega$ , but the  $\beta$ -nf of the term is  $\lambda y.y$

*Proof.* Denote  $\beta$ -nf of  $M$  as  $M'$ . For any form in the reduction path,  $M \xrightarrow{\beta} M_i$ . In conjunction with  $M \xrightarrow{\beta} M'$ , by the Church-Rosser theorem, there exists  $L$  such that  $M_i \xrightarrow{\beta} L$  and  $M' \xrightarrow{\beta} L$ . Because  $M'$  is in  $\beta$ -nf,  $L$  can only be  $M'$ , thus  $M_i \xrightarrow{\beta} M'$ , so  $M_i$  is capable of reducing to  $M'$ , a  $\beta$ -nf. Therefore, any form in the reduction path has a  $\beta$ -nf. ■

### Problem

(1.17) If  $MN$  is strongly normalizing, then both  $M$  and  $N$  are strongly normalizing.

*Solution.*

*Proof.* If  $M$  is not strongly normalizing, then there exists a reduction path  $M_0 \xrightarrow{\beta} M_1 \xrightarrow{\beta} \dots$ . Therefore,  $MN$  would have had a reduction path  $MN \xrightarrow{\beta} M_1N \xrightarrow{\beta} \dots$  that is infinite, which contradicts with  $MN$  being strongly normalizing. Vice versa for  $N$ . ■

### Problem

(1.18) Let  $L, M, N \in \Lambda$  such that  $L \equiv_{\beta} M$  and  $L \xrightarrow{\beta} N$ . Moreover,  $N$  is in  $\beta$ -nf. Prove that  $M \xrightarrow{\beta} N$ .

*Solution.* Corollary 1.9.9.

### Problem

(1.19) Define

$$U := \lambda zx.x(zzx) \quad \text{and} \quad Z := UU$$

Prove  $Z$  is a fixed point combinator.

*Solution.* Proving  $\forall L \in \Lambda, L(ZL) \xrightarrow[\beta]{} ZL$ .

*Proof.*

$$\begin{aligned} ZL &= (\lambda zx.x(zzx))(\lambda zx.x(zzx))L \\ &\xrightarrow[\beta]{} L((\lambda zx.x(zzx))(\lambda zx.x(zzx))L) \xrightarrow[\beta]{} L(ZL) \end{aligned}$$

■

### Problem

(1.20) Solve for  $M \in \Lambda$  in each equation:

$$\begin{aligned} M &\stackrel{=}{\beta} \lambda xy.xMy \\ Mxyz &\stackrel{=}{\beta} xyzM \end{aligned}$$

*Solution.* By the property of the  $Y$  combinator:

$$f(Yf) = Yf$$

The first equation can be remodeled as

$$M \stackrel{=}{\beta} LM \quad \text{where} \quad L = \lambda mxy.xmy$$

Solving for fixed point of  $L$  :

$$\begin{aligned} M &\equiv YL \equiv L(YL) \\ &= (\lambda x.L(xx))(\lambda x.L(xx)) \\ &= (\lambda x.(\lambda muv.umv)(xx))(\lambda x.(\lambda muv.umv)(xx)) \\ &\xrightarrow[\beta]{} (\lambda x.(\lambda uv.u(xx)v))(\lambda x.(\lambda uv.u(xx)v)) \end{aligned}$$

The second equation can be  $\eta$ -reduced on both sides:

$$\begin{aligned} Mxyz &\stackrel{=}{\beta} xyzM \\ M &\stackrel{=}{\beta} \lambda xyz.xyzM \end{aligned}$$

Remodeling equation:

$$M = NM \text{ where } M = \lambda mxyz.xyzm$$

then

$$\begin{aligned} M &\equiv YN \equiv N(YN) \\ &= (\lambda x.N(xx))(\lambda x.N(xx)) \\ &= (\lambda x.(\lambda mxyz.xyzm)(xx))(\lambda x.(\lambda mxyz.xyzm)(xx)) \\ &\xrightarrow[\beta]{} (\lambda x.(\lambda yz.xyz(xx)))(\lambda x.(\lambda yz.xyz(xx))) \end{aligned}$$