

CMSC330 Fall 2013 Practice Problems 8 Solutions

1. Lambda calculus

Make all parentheses explicit in the following λ -expressions

- a. $\lambda x.xz \lambda y.xy \rightarrow (\lambda x.((x z) (\lambda y.(x y))))$
 b. $(\lambda x.xz) \lambda y.w \lambda w.wyzx \rightarrow ((\lambda x.(x z)) (\lambda y.(w (\lambda w.((((w y) z) x))))))$
 c. $\lambda x.xy \lambda x.yx \rightarrow (\lambda x.((x y) (\lambda x.(y x))))$

Find all free (unbound) variables in the following λ -expressions

- d. $\lambda x.x z \lambda y.x y \rightarrow (\lambda x.((x \underline{z}) (\lambda y.(x y))))$
 e. $(\lambda x.x z) \lambda y.w \lambda w.w y z x \rightarrow ((\lambda x.(x \underline{z})) (\lambda y.(\underline{w} (\lambda w.((((w y) \underline{z}) \underline{x}))))))$
 f. $\lambda x.x y \lambda x.y x \rightarrow (\lambda x.((x \underline{y}) (\lambda x.(\underline{y} x))))$

Apply β -reduction to the following λ -expressions as much as possible

- g. $(\lambda z.z) (\lambda y.y y) (\lambda x.x a) \rightarrow$ // β -reduction = body[sym/replacement]
 $(\lambda z.\underline{z}) (\lambda y.\underline{y} y) (\lambda x.\underline{x} a) \rightarrow$ // $z[z/(\lambda y.y y)]$ replace $\underline{z}$ with $\lambda y.y y$
 $(\lambda y.\underline{y} y) (\lambda x.\underline{x} a) \rightarrow$ // $y y[y/(\lambda x.x a)]$ replace $\underline{y}$ with $\lambda x.x a$
 $(\lambda x.\underline{x} a) (\lambda x.\underline{x} a) \rightarrow$ // $x a[x/(\lambda x.x a)]$ replace $\underline{x}$ with $\lambda x.x a$
 $(\lambda x.\underline{x} a) a \rightarrow a a$ // $x a[x/a]$ replace $\underline{x}$ with a
- h. $(\lambda z.z) (\lambda z.z z) (\lambda z.z y) \rightarrow$ // β -reduction: replace $\underline{z}$ with $\lambda z.z z$
 $(\lambda z.\underline{z}) (\lambda z.\underline{z} z) (\lambda z.\underline{z} y) \rightarrow$ // β -reduction: replace $\underline{z}$ with $\lambda z.z y$
 $(\lambda z.\underline{z} y) (\lambda z.\underline{z} y) \rightarrow$ // β -reduction: replace $\underline{z}$ with $\lambda z.z y$
 $(\lambda z.\underline{z} y) y \rightarrow y y$ // β -reduction: replace $\underline{z}$ with y
- i. $(\lambda x.\lambda y.x y y) (\lambda a.a) b \rightarrow$ // β -reduction: replace $\underline{x}$ with $\lambda a.a$
 $(\lambda x.\lambda y.\underline{x} y y) (\lambda a.\underline{a}) b \rightarrow$ // β -reduction: replace $\underline{y}$ with b
 $(\lambda a.\underline{a}) b b \rightarrow b b$ // β -reduction: replace $\underline{a}$ with b
- j. $(\lambda x.\lambda y.x y y) (\lambda y.y) y \rightarrow$ // α -conversion: rename $\underline{y}$ to a
 $(\lambda x.\lambda y.\underline{x} y y) (\lambda y.\underline{y}) y \rightarrow$ // β -reduction: replacing $\underline{x}$ with $\lambda y.y$
 $(\lambda a.(\lambda y.y) \underline{a} \underline{a}) y \rightarrow$ // β -reduction: replacing $\underline{a}$ with y
 $(\lambda y.\underline{y}) y y \rightarrow y y$ // β -reduction: replacing $\underline{y}$ with y
- k. $(\lambda x.x x) (\lambda y.y x) z \rightarrow$ // β -reduction: replacing $\underline{x}$ with $\lambda y.y x$
 $(\lambda y.\underline{y} x) (\lambda y.\underline{y} x) z \rightarrow$ // β -reduction: replacing $\underline{y}$ with $\lambda y.y x$
 $(\lambda y.\underline{y} x) x z \rightarrow$ // β -reduction: replacing $\underline{y}$ with x
 $x x z$
- l. $(\lambda x.(\lambda y.(x y)) y) z \rightarrow$ // α -conversion: rename $\underline{y}$ to a
 $(\lambda x.(\lambda y.(\underline{x} \underline{y})) y) z \rightarrow$ // β -reduction: replacing $\underline{x}$ with z
 $(\lambda x.(\lambda a.(\underline{x} a)) y) z \rightarrow$ // β -reduction: replacing $\underline{a}$ with y
 $(\lambda a.(z \underline{a})) y \rightarrow$
 $z y$

- m. $((\lambda x. x \ x) (\lambda y. y)) (\lambda y. y)$
 $((\lambda x. \mathbf{x} \ \mathbf{x}) (\lambda \mathbf{y}. \mathbf{y})) (\lambda y. y) \rightarrow$ // β -reduction: replacing $\mathbf{x}$ with $\lambda y. y$
 $((\lambda y. \mathbf{y}) (\lambda \mathbf{y}. \mathbf{y})) (\lambda y. y) \rightarrow$ // β -reduction: replacing $\mathbf{y}$ with $\lambda y. y$
 $(\lambda y. y) (\lambda \mathbf{y}. \mathbf{y}) \rightarrow$ // β -reduction: replacing $\mathbf{y}$ with $\lambda y. y$
 $\lambda y. y$
- n. $((\lambda x. \lambda y. (x \ y)) (\lambda y. y)) w$
 $((\lambda x. \lambda \mathbf{y}. (x \ \mathbf{y})) (\lambda y. y)) w \rightarrow$ // α -conversion: rename $\mathbf{y}$ to a
 $((\lambda x. \lambda a. (\mathbf{x} \ a)) (\lambda \mathbf{y}. \mathbf{y})) w \rightarrow$ // β -reduction: replacing $\mathbf{x}$ with $\lambda y. y$
 $((\lambda a. ((\lambda y. y) \ \mathbf{a})) w) \rightarrow$ // β -reduction: replacing $\mathbf{a}$ with w
 $(\lambda y. \mathbf{y}) \ \mathbf{w} \rightarrow$ // β -reduction: replacing $\mathbf{y}$ with $\lambda y. y$
 w

Show that the following expression has multiple reduction sequences

- o. $(\lambda x. y) ((\lambda y. y \ y \ y) (\lambda x. x \ x \ x))$
// β -reduction: replace x in $\lambda x. y$ with $((\lambda y. y \ y \ y) (\lambda x. x \ x \ x))$
// (no x in body, so just discard argument and replace $(\lambda x. y) <...>$ with y)
 $(\lambda x. y) ((\lambda y. y \ y \ y) (\lambda \mathbf{x}. \mathbf{x} \ \mathbf{x} \ \mathbf{x})) \rightarrow y$
OR
// β -reduction: replace y in $\lambda y. y \ y \ y$ with $\lambda x. x \ x \ x$
 $(\lambda x. y) ((\lambda y. \mathbf{y} \ \mathbf{y} \ \mathbf{y}) (\lambda \mathbf{x}. \mathbf{x} \ \mathbf{x} \ \mathbf{x})) \rightarrow (\lambda x. y) ((\lambda x. x \ x \ x) (\lambda x. x \ x \ x) (\lambda x. x \ x \ x))$

Can repeat β -reduction for x as many times as we wish!

2. Lambda calculus encodings

Prove the following using the appropriate λ -calculus encodings

- a. $\text{not} (\text{not true}) = \text{true}$

Given:

$\text{not} = \lambda x. ((x \ \text{false}) \ \text{true})$

$\text{true} = \lambda x. \lambda y. x$

$\text{false} = \lambda x. \lambda y. y$

Proof:

$\text{not} (\text{not true})$	// replacing 1 st not w/ encoding
$= \lambda x. ((\mathbf{x} \ \text{false}) \ \text{true}) (\text{not true})$	// β -reduction: $\mathbf{x} \rightarrow \text{not true}$
$= ((\text{not true}) \ \text{false}) \ \text{true}$	// replacing not w/ encoding
$= ((\lambda x. ((\mathbf{x} \ \text{false}) \ \text{true}) \ \mathbf{true}) \ \text{false}) \ \text{true}$	// β -reduction: $\mathbf{x} \rightarrow \text{true}$
$= (((\text{true} \ \text{false}) \ \text{true}) \ \text{false}) \ \text{true}$	// replace true w/ encoding
$= (((\lambda x. \lambda y. \mathbf{x}) \ \text{false}) \ \text{true}) \ \text{false}) \ \text{true}$	// β -reduction: 1 st $\mathbf{x} \rightarrow \text{false}$
$= (((\lambda y. \text{false}) \ \mathbf{true}) \ \text{false}) \ \text{true}$	// β -reduction: $\mathbf{y} \rightarrow \text{true}$
$= ((\text{false}) \ \text{false}) \ \text{true}$	// replace false w/ encoding
$= ((\lambda x. \lambda y. y) \ \text{false}) \ \text{true}$	// β -reduction: $\mathbf{x} \rightarrow \text{false}$
$= (\lambda y. \mathbf{y}) \ \text{true}$	// β -reduction: $\mathbf{y} \rightarrow \text{true}$
$= \text{true}$	// $\text{not} (\text{not true}) = \text{true}$

b. or false true = true

Given:

or = $\lambda x. \lambda y. ((x \text{ true}) y)$

true = $\lambda x. \lambda y. x$

false = $\lambda x. \lambda y. y$

Proof:

or false true

= $\lambda x. \lambda y. ((x \text{ true}) y) \text{ false true}$

= $\lambda y. ((\text{false true}) y) \text{ true}$

= $(\text{false true}) \text{ true}$

= $((\lambda x. \lambda y. y) \text{ false}) \text{ true}$

= $(\lambda y. y) \text{ true}$

= true

// replacing or w/ encoding

// β -reduction: $x \rightarrow \text{false}$

// β -reduction: $y \rightarrow \text{true}$

// replace 1st false w/ encoding

// β -reduction: $x \rightarrow \text{false}$

// β -reduction: $y \rightarrow \text{true}$

// or false true = true

c. if false then x else y = y

Given:

if a then b else c = $\lambda a b c$

true = $\lambda x. \lambda y. x$

false = $\lambda x. \lambda y. y$

Proof:

if false then x else y

= false x y

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

= $(\lambda y. y) y$

= y

// replacing if... w/ encoding

// replacing false w/ encoding

// β -reduction: $x \rightarrow x$

// β -reduction: $y \rightarrow y$

// if false then x else y = y

d. succ 2 = 3

Given:

2 = $\lambda f. \lambda y. f (f y)$

3 = $\lambda f. \lambda y. f (f (f y))$

succ = $\lambda z. \lambda f. \lambda y. f (z f y)$

Proof:

succ 2

= $(\lambda z. \lambda f. \lambda y. f (z f y)) 2$

= $\lambda f. \lambda y. f (2 f y)$

= $\lambda f. \lambda y. f ((\lambda f. \lambda y. f (f y)) f y)$

= $\lambda f. \lambda y. f ((\lambda y. f (f y)) y)$

= $\lambda f. \lambda y. f (f (f y))$

= 3

// replacing succ w/ encoding

// β -reduction: $z \rightarrow 2$

// expanding 2 w/ encoding

// β -reduction: 1st f $\rightarrow f$

// β -reduction: 1st y $\rightarrow y$

// apply encoding for 3

// succ 2 = 3

e. (* 1 3) = 3

Given:

1 = $\lambda f. \lambda y. f y$

3 = $\lambda f. \lambda y. f (f (f y))$

M * N = $\lambda x. (M (N x))$

Proof:

(* 1 3)

= $\lambda x. (1 (3 x))$

// replacing * w/ encoding

// replacing 3 w/ encoding

$$\begin{aligned}
&= \lambda x.(1 (\lambda f.\lambda y.f (f (f y)) x)) && // \beta\text{-reduction: } 1^{\text{st}} f \rightarrow x \\
&= \lambda x.(1 (\lambda y.x (x (x y)))) && // \text{replacing } 1 \text{ w/ encoding} \\
&= \lambda x.((\lambda f.\lambda y.f y) (\lambda y.x (x (x y)))) && // \beta\text{-reduction: } 1^{\text{st}} f \text{ w/ } \lambda y.x (x (x y)) \\
&= \lambda x.(\lambda y.(\lambda y.x (x (x y)))) y && // \beta\text{-reduction: } 1^{\text{st}} y \rightarrow y \\
&= \lambda x.\lambda y.x (x (x y)) && // \alpha\text{-conversion: replace } x \text{ with } f \\
&= \lambda f.\lambda y.f (f (f y)) && // \text{apply encoding for } 3 \\
&= 3
\end{aligned}$$

f. $(+ 2 1) = 3$

Given:

$$\begin{aligned}
1 &= \lambda f.\lambda y.f y \\
2 &= \lambda f.\lambda y.f (f y) \\
3 &= \lambda f.\lambda y.f (f (f y)) \\
M + N &= \lambda x.\lambda y.(M x)((N x) y)
\end{aligned}$$

Proof:

$$\begin{aligned}
&(+ 2 1) && // \text{replacing } + \text{ w/ encoding} \\
&= \lambda x.\lambda y.(2 x)((1 x) y) && // \text{replacing } 2 \text{ w/ encoding} \\
&= \lambda x.\lambda y.((\lambda f.\lambda y.f (f y)) x)((1 x) y) && // \beta\text{-reduction: } 1^{\text{st}} f \rightarrow x \\
&= \lambda x.\lambda y.(\lambda y.x (x y))((1 x) y) && // \text{replacing } 1 \text{ w/ encoding} \\
&= \lambda x.\lambda y.(\lambda y.x (x y))(((\lambda f.\lambda y.f y) x) y) && // \beta\text{-reduction: } 1^{\text{st}} f \rightarrow x \\
&= \lambda x.\lambda y.(\lambda y.x (x y))((\lambda y.x y) y) && // \beta\text{-reduction: } 3^{\text{rd}} y \rightarrow y \\
&= \lambda x.\lambda y.(\lambda y.x (x y))(x y) && // \beta\text{-reduction: } 2^{\text{nd}} y \rightarrow x y \\
&= \lambda x.\lambda y.x (x (x y)) && // \alpha\text{-conversion: replace } x \text{ with } f \\
&= \lambda f.\lambda y.f (f (f y)) && // \text{apply encoding for } 3 \\
&= 3
\end{aligned}$$

g. $(Y \text{ fact}) 2 = 2$ // you do not need to expand any operators except fact & Y

Given:

$$\begin{aligned}
Y &= \lambda f.(\lambda x.f (x x)) (\lambda x.f (x x)) \\
\text{fact} &= \lambda f. \lambda n.\text{if } n = 0 \text{ then } 1 \text{ else } n * (f (n-1))
\end{aligned}$$

Proof:

$$\begin{aligned}
&(Y \text{ fact}) 2 && // \text{replacing } Y \text{ w/ encoding} \\
&= (\lambda f.(\lambda x.f (x x)) (\lambda x.f (x x))) \text{fact} 2 && // \beta\text{-reduction: } 1^{\text{st}} f \rightarrow \text{fact} \\
&= (\lambda x.\text{fact} (x x)) (\lambda x.\text{fact} (x x)) 2 && // \beta\text{-reduction: } 1^{\text{st}} x \rightarrow \lambda x.\text{fact} (x x) \\
&= (\text{fact} ((\lambda x.\text{fact} (x x)) (\lambda x.\text{fact} (x x)))) 2 \\
&\quad // \text{apply encoding for } (Y \text{ fact}) \\
&\quad // ((\lambda x.\text{fact} (x x)) (\lambda x.\text{fact} (x x))) \rightarrow (Y \text{ fact}) \\
&\quad // \text{we know this is the encoding for } (Y \text{ fact}) \text{ from } 3^{\text{rd}} \text{ line of proof} \\
&= (\text{fact} (Y \text{ fact})) 2 && // \text{apply encoding for fact} \\
&= (\lambda f. \lambda n.\text{if } n = 0 \text{ then } 1 \text{ else } n * (f (n-1))) (Y \text{ fact}) 2 \\
&\quad // \beta\text{-reduction: } 1^{\text{st}} f \rightarrow (Y \text{ fact}) \\
&= (\lambda n.\text{if } n = 0 \text{ then } 1 \text{ else } n * ((Y \text{ fact}) (n-1))) 2 && // \beta\text{-reduction: } n \rightarrow 2 \\
&= \text{if } 2=0 \text{ then } 1 \text{ else } 2 * ((Y \text{ fact}) (2-1)) && // \text{apply if} \\
&= 2 * ((Y \text{ fact}) 1) && // \text{showed in class } (Y \text{ fact}) 1 = 1 \\
&= 2 * 1 && // \text{apply } * \\
&= 2
\end{aligned}$$