1. Lambda calculus

Make all parentheses explicit in the following λ-expressions

a. λx.xz λy.xy **(λx.((x z) (λy.(x y))))**

b. (λx.xz) λy.w λw.wyzx **((λx.(x z)) (λy.(w (λw.((((w y) z) x))))))**

c. λx.xy λx.yx **(λx.((x y) (λx.(y x))))**

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

d. λx.x z λy.x y **(λx.((x z) (λy.(x y))))**

e. (λx. x z) λy. w λw. w y z x **((λx.(x z)) (λy.(w (λw.((((w y) z) x))))))**

f. λx. x y λx. y x **(λx.((x y) (λx.(y x))))**

3.

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

g. (λz.z) (λy.y y) (λx.x a) // **β-reduction = body[sym/replacement]**

(λz.z) (λy.y y) (λx.x a) // z[z/(λy.y y)] replace z with λy.y y

(λy.y y) (λx.x a) // y y[y/(λx.x a)] replace y with λx.x a

(λx.x a) (λx.x a) // x a[x/(λx.x a)] replace x with λx.x a

(λx.x a) a // x a[x/a] replace x with a

a a

h. (λz.z) (λz.z z) (λz.z y)

(λz.z) (λz.z z) (λz.z y) _ // β-reduction: replace z with λz.z z

(λz.z z) (λz.z y) _ // β-reduction: replace z with λz.z y

(λz.z y) (λz.z y) _ // β-reduction: replace z with λz.z y

(λz.z y) y _ // β-reduction: replace z with y

y y

i. (λx.λy.x y y) (λa.a) b

(λx.λy.x y y) (λa.a) b _ // β-reduction: replace x with λa.a

(λy.(λa.a) y y) b _ // β-reduction: replace y with b

(λa.a) b b _ // β-reduction: replace a with b

b b

j. (λx.λy.x y y) (λy.y) y

(λx.λy.x y y) (λy.y) y _ // α-conversion: rename y to a

(λx.λa.x a a) (λy.y) y _ // β-reduction: replacing x with λy.y

(λa.(λy.y) a a) y _ // β-reduction: replacing a with y

(λy.y) y y _ // β-reduction: replacing y with y

y y

k. (λx.x x) (λy.y x) z

(λx.x x) (λy.y x) z _ // β-reduction: replacing x with λy.y x

(λy.y x) (λy.y x) z _ // β-reduction: replacing y with λy.y x

(λy.y x) x z _ // β-reduction: replacing y with x

x x z

l. (λx. (λy. (x y)) y) z

(λx. (λy. (x y)) y) z _ // α-conversion: rename y to a

(λx. (λa. (x a)) y) z _ // β-reduction: replacing x with z

(λa. (z a)) y _ // β-reduction: replacing a with y

z y

m. ((λx.x x) (λy.y)) (λy.y)

((λx.x x) (λy.y)) (λy.y) _ // β-reduction: replacing x with λy.y

((λy.y) (λy.y)) (λy.y) _ // β-reduction: replacing y with λy.y

(λy.y) (λy.y) _ // β-reduction: replacing y with λy.y

λy.y

n. (((λx. λy.(x y))(λy.y)) w)

(((λx. λy.(x y))(λy.y)) w) _ // α-conversion: rename y to a

(((λx. λa.(x a))(λy.y)) w) _ // β-reduction: replacing x with λy.y

((λa.((λy.y) a)) w) _ // β-reduction: replacing a with w

(λy.y) w _ // β-reduction: replacing y with λy.y

w