Lambda Calculus

For the following forms apply β-reduction and α-substitution to reduce to lowest form. Indicate at each step the rule you are applying.

1. (λx.x)( λx.x )

=> λx.x -- β-reduction where x = ( λx.x )

1. (λx.x x) (λx. λy.x x)

=> (λx.(x x)) (λx.(λy.x x)) -- Application is left associative

=> ((λx.(λy.x x)) (λx.(λy.x x))) -- β-reduction where x = (λx.(λy.x x))

=> λy. (λx.(λy.x x)) ((λx.(λy.x x)) -- β-reduction where x = (λx.(λy.x x))

1. ((λx.(x y))(λz.z))

=> ((λz.z)y) -- β-reduction where x = (λz.z))

=> y -- β-reduction where z = y

1. (λz.z) (λy.y y) (λx.x a)

=> ((λz.z) (λy.y y)) (λx.x a) -- Application is left associative

=> (λy.y y) (λx.x a) -- β-reduction where z = (λy.y y))

=> (λx.x a) (λx.x a) -- β-reduction where y = (λx.x a)

=> ((λx.x a) a) -- β-reduction where x = (λx.x a)

=> (a a) -- β-reduction where x = a

1. (λz.z) (λz.z z) (λz.z y)

=> ((λz.z) (λc.c c)) (λz.z y) -- α-substitution, z in expression (λz.z z) substituted to c

=> (λc.c c) (λz.z y) -- β-reduction where z=(λc.c c)

=> (λc.c c) (λk.k y) -- α-substitution, z in expression (λz.z y) substituted to k

=> (λk.k y) (λk.k y) -- β-reduction where c= (λk.k y)

=> ((λk.k y)y) -- β-reduction where k= (λk.k y)

=> (y y) -- β-reduction where k=y

1. (λx.λy.x y y) (λa.a) b

=> (λy.( λa.a) y y)b -- β-reduction where x=( λa.a)

=> ( λa.a)b b -- β-reduction where y=b

=> (b b) -- β-reduction where a=b b

1. (λx.x x) (λy.y x) z

=> ((λx.x x) (λy.y c)) z --- α-substitution, x in expression (λy.y x) substituted to c

=> ((λy.y c) (λy.y c)) z -- β-reduction where x=(λy.y c)

=> ((λy.y c)c) z -- β-reduction where y=(λy.y c)

=> (cc)z -- β-reduction where y=c

1. (λx. (λy. (x y)) y) z

=> (λx. (λy. (x y)) y) z

=> (λx. (λy. (x y)) t) z --- α-substitution, right most y in expression (λx. (λy. (x y)) y) substituted to t as it is not bound

=> (λy. (z y)) t --- β-reduction where x=z

=> z t --- β-reduction where y=t

1. ((λx.x x) (λy.y)) (λy.y)

* ((λy.y) (λy.y)) (λy.y) --- β-reduction where x=(λy.y)
* ((λy.y) (λy.y)) --- β-reduction where y=(λy.y)
* ((λy.y) (λt.t)) --- α-substitution, right most y in expression (λy.y) substitute to t
* (λt.t) --- β-reduction where y=(λt.t)

1. (((λx. λy.(x y))(λy.y)) w)

* (λy.((λy.y) y))w --- β-reduction where x=(λy.y)
* (λy.(λt.t)y)w --- α-substitution, right most y in expression (λy.((λy.y) y))w substituted to t
* (λt.t)w --- β-reduction where y=w
* w --- β-reduction where t=w