Aufgabe 1.

and = (λx. λy. if x y False)
or = (λx λy. if x True y)
not = (λx. if x False True

and True False:

and = (λx. λy. if x y False) True False
= λy. if True y False
= if True False False
= False

or False True:

or = (λx. λy if x True y) False True
= (λy if False True y) True
= if False True True
= True

Aufgabe 2.

1) (λx. λy · 5 x) y

β-Reduktion, kein α-Konversion, dennes ist einfach lösbar, wenn wir Argument für y haben

(λx. λy. · 5 x) y = (λy. · 5 y)

2) (λx. if (= x 0) 42 x) (- 5 4)

β-Reduktion, kein α-Konversion, denn einfach lösbar

(if (= (-5) 0) 42 (-5)) (4)

-5 ≠ 0 => 4.

3) (λf x. f (f x)) (λx. x × 8)

β-Reduktion

λx. (λx. x × 8) ((λx. x × 8) x)

α-Konversion x = y, einfacher zu verstehen

λx. (λy × y 8) ((λ. y × y 8) y)

4) (λf a x. f x a) (λf g. (λx. f (g x)))

β-Reduktion,

⇒ (λ a x. (λf g. (λx. f (g x))) a)

⇒ (λ a x. (λf g. (λx. f (g x)) a)

⇒ λ a x. (λf g. (λx. f (g x) a))

⇒ λ a x. (λf g. f (g a))

α-Konversion

⇒ λ a x. (λ g. g a)

5) (λf x. g (λx f x)) x (λx. x)

$\xrightarrow{\beta}$ (λf x. g (λx (λx. x) x)) x

$\xrightarrow{\beta}$ (λf x. g x) x

$\xrightarrow{\beta}$ (λf x. g x) ???

Aufgabe 4.

3. Die Anzahl der Multiplikation, die zur Berechnung von nat-pow x k benötigt werden, entspricht der Quersumme von k in Binärsystem, also crossSum 2 k.