reductions):
XOR p (NOT p). For all computations you perform, indicate clearly the reduction being done by underlying the abstraction used and the argument it is applied to: (λx.M)N.
(b) (6pt) Does your computation indicate that XOR is behaving as the exclusive-or operator you know?

(a) Call-by-value reduction:

XOR p (NOT p)
= (λpq.p (NOT q) q) p (NOT p) [by substitution] => λq. p (NOT q) q
= p (NOT (NOT p)) (NOT p) [by reduction]
= p (λqr.r p q) (NOT p) [by reduction]
= (λpq.p) (λqr.r p q) (NOT p) [by substitution]
= (λqr.r p q) (NOT p) [by reduction]
= (λr.NOT p r p) [by reduction]

Call-by-name reduction:

XOR p (NOT p)
= (λpq.p (NOT q) q) p (NOT p) [by substitution]
= p (NOT (NOT p)) (NOT p) [by reduction]
= p (λr.NOT p r p) (NOT p) [by reduction]
= (λp.(λr.NOT p r p) p (NOT p)) [by abstraction]
= (λr.NOT p r p) (NOT p) [by reduction]
= (λp.(λr.NOT p r p) (NOT p)) [by abstraction]
= (λr.NOT (NOT p) r (NOT p)) [by reduction]

(b) Yes, the computation indicates that XOR behaves as the exclusive-or operator, as the result is p if the values of p and (NOT p) are different, and (NOT p) if they are the same, which is the definition of exclusive-or.

2 / 2 >