

<pre> true = \x.\y.x false = \x.\y.y  or = \p.\q.((p q) false) and = \p.\q.((p true) q) not = \p.\a.\b.((p b) a) if = \b.\t.\e.((b t) e) = \b.b xor = \p.\q.((p (q false true)) q) eq = \p.\q.(not ((xor p) q)) </pre>	<pre> 0 = \f.\x.x 1 = \f.\x.(f x) 2 = \f.\x.(f (f x)) 3 = \f.\x.(f (f (f x))) ... n = \f.\x.(f ... (f x) ... ) </pre>
<p>Alpha Renaming: <math>(\lambda y (y (y w))) \rightarrow (\lambda g (g (y w)))</math></p> <p>Beta Reduction: <math>(\lambda p. B a) \rightarrow</math> plug in a for all instances of p in B</p> <p>Eta Reduction: <math>(\lambda x. (E x)) \rightarrow E</math></p>	<pre> succ = \n.\f.\x.(f ((n f) x)) add = \n.\m.((m succ) n) mult = \n.\m.\f.(m (n f)) exp = \n.\m.(n m) </pre>
	<p><b>Pairs:</b></p> <pre> pr = \x.\y.\b.((b x) y) 1st = \l.(l true) 2nd = \l.(l false) </pre>
	<p><b>List:</b></p> <pre> nil = \l.true prepend = pr head = 1st tail = 2nd isNil? = \l.(l \h.\t.false) </pre>