variable data : Type

variables

L15\_E R15\_E L16\_E R16\_E

L0\_D R0\_D L1\_D R1\_D

K\_16 : data

variable xor (d1 d2 : data) : data

variable f (d k : data) : data

premise xor\_assoc (x1 x2 x3 : data) :

(xor (xor x1 x2) x3) = (xor x1 (xor x2 x3))

premise xor\_cancel : ∀ x1 x2 : data, xor x1 (xor x2 x2) = x1

premise H1 : L16\_E = R15\_E

premise H2 : R16\_E = xor L15\_E (f R15\_E K\_16)

premise H3 : L0\_D = R16\_E

premise H4 : R0\_D = L16\_E

premise H5 : L1\_D = R0\_D

premise H6 : R1\_D = xor L0\_D (f R0\_D K\_16)

theorem ProofRight\_Long : R15\_E = L1\_D :=

have H7 : R15\_E = L16\_E, from eq.symm H1,

have H8 : L16\_E = R0\_D, from eq.symm H4,

have H9 : R0\_D = L1\_D, from eq.symm H5,

have H10 : R15\_E = R0\_D, from eq.trans H7 H8,

show R15\_E = L1\_D, from eq.trans H10 H9

theorem ProofLeft : L15\_E = R1\_D :=

have H11 : R1\_D = xor L0\_D (f R0\_D K\_16), from H6,

have H12 : R1\_D = xor R16\_E (f R0\_D K\_16), from eq.subst H3 H11,

have H13 : R1\_D = xor R16\_E (f L16\_E K\_16), from eq.subst H4 H12,

have H14 : R1\_D = xor R16\_E (f R15\_E K\_16), from eq.subst H1 H13,

have H15 : R1\_D = xor (xor L15\_E (f R15\_E K\_16)) (f R15\_E K\_16),

from eq.subst H2 H14,

have H16 : xor (xor L15\_E (f R15\_E K\_16)) (f R15\_E K\_16) =

xor L15\_E (xor (f R15\_E K\_16) (f R15\_E K\_16)),

from xor\_assoc L15\_E (f R15\_E K\_16) (f R15\_E K\_16),

have H17 : R1\_D = xor L15\_E (xor (f R15\_E K\_16) (f R15\_E K\_16)),

from eq.trans H15 H16,

have H18 : xor L15\_E (xor (f R15\_E K\_16) (f R15\_E K\_16)) = L15\_E,

from xor\_cancel L15\_E (f R15\_E K\_16),

show L15\_E = R1\_D, from eq.symm (eq.trans H17 H18)