Aufgabe 7-2

Zu zeigen: (reverse (append l1 l2)) ≡ (append (reverse l2) (reverse l1))

Basisfall: l1 ist leer

LHS (Linke Hand Seite)

≡ (reverse (append empty l2))

Aussage 5.3a, EKONG

≡ (reverse l2)

RHS (Rechte Hand Seite)

≡ (append (reverse l2) (reverse empty))

EFUN, EKONG

≡ (append (reverse l2) (cond [(empty? x) empty]
[…]))

PRIM, ERED, EKONG

≡ (append (reverse l2) (cond [#true empty]
[…]))

COND-true, ERED

≡ (append (reverse l2) empty)

Aussage 5.3b, EKONG

≡ (reverse l2)

ETRANS => LHS ≡ RHS ≡ (reverse l2)

Induktionsannahme:

Sei x beliebig, l1 ≡ (cons x l0)

(reverse (append l0 l2)) ≡ (append (reverse l2) (reverse l0))

Zu zeigen (reverse (append l1 l2)) ≡ (append (reverse l2) (reverse l1))

Es gilt: (reverse l1)

EFUN, EKONG

≡ (cond [(empty? (cons x l0)) empty]
[(cons? (cons x l0)) (append (reverse (rest (cons x l0)))
(cons (first (cons x l0)) empty))])

PRIM, ERED, EKONG

≡ (cond [#false empty]
[(cons? (cons x l0)) (append (reverse (rest (cons x l0)))
(cons (first (cons x l0)) empty))])

COND-false, ERED

≡ (cond [(cons? (cons x l0)) (append (reverse (rest (cons x l0)))
(cons (first (cons x l0)) empty))])

STRUCT-predtrue, EKONG

≡ (cond [#true (append (reverse (rest (cons x l0)))
(cons (first (cons x l0)) empty))])

COND-true, ERED

≡ (append (reverse (rest (cons x l0))) (cons (first (cons x l0)) empty))

PRIM, ERED, EKONG

≡ (append (reverse l0) (cons x empty)) <*>

Induktionschritt:

LHS

EKONG, Induktionsannahme

≡ (reverse (append (cons x l0) l2))

Aussage 5.3c, EKONG

≡ (reverse (cons x (append l0 l2))

EFUN, EKONG

≡ (cond [(empty? (cons x (append l0 l2)) empty]
[(cons? (cons x (append l0 l2)) (append (reverse (rest (cons x (append l0 l2)))
(cons (first (cons x (append l0 l2)) empty))]))

PRIM, ERED, EKONG

≡ (cond [#false empty]
[(cons? (cons x (append l0 l2)) (append (reverse (rest (cons x (append l0 l2)))
(cons (first (cons x (append l0 l2)) empty))]))

COND-false, ERED

≡ (cond [(cons? (cons x (append l0 l2)) (append (reverse (rest (cons x (append l0 l2)))
(cons (first (cons x (append l0 l2)) empty))]))

STRUCT-predtrue, EKONG

≡ (cond [#true (append (reverse (rest (cons x (append l0 l2)))
(cons (first (cons x (append l0 l2)) empty))]))

COND-true, ERED

≡ (append (reverse (rest (cons x (append l0 l2)))(cons (first (cons x (append l0 l2)) empty))
PRIM, ERED, EKONG
≡ (append (reverse (append l0 l2)))(cons x empty))
Induktionsannahme, EKONG
≡ (append (append (reverse l2) (reverse l0)))(cons x empty))
Aussage 5.3d, EKONG
≡ (append (reverse l2) (append (reverse l0)(cons x empty)))
EKONG, <*>
≡ (append (reverse l2) (reverse l1)) (RHS)

Also (reverse (append l1 l2)) ≡ (append (reverse l2) (reverse l1)) für alle Listen l1 und l2.