HW2 - Q2

Proof: rev_append' l1 l2 = append(rev l_1) l_2 = rev_append $l_1 l_2$

<u>Base</u> l_1 = empty list

1) rev_append' [ ] $l_2 \underset{\text{by def.}}{=}$ append (rev [ ]) $l_2 \overset{\text{by rev}}{\Longrightarrow}$ append [ ] $l_2 \Downarrow l_2$

2) rev_append [ ] $l_2 \underset{\text{by rev\_append}}{\Downarrow} l_2$

→ Both methods evaluate to 'l_2'

<u>case</u>: $l = h::t$

<u>IH</u>: For $l_1 = h::t$ rev_append' t l_2 = rev_append t l_2

To show rev_append' l_1 l_2 = rev_append l_1 l_2

Assume: append l_1 l_2 = l_1 @ l_2

rev_append' l_1 $l_2 \underset{\text{by def of rev\_append'}}{=}$ append (rev l1) l2

$\underset{\text{by rev}}{\Longrightarrow}$ append [(rev t) @ [h] l2]

$\underset{\text{by assumption}}{\Longrightarrow}$ [rev t @ [h]] @ l_2

$\underset{\text{by associativity}}{\Longrightarrow}$ (rev t) @ [[h] @ l_2] $\underset{\text{represent @ as (::)}}{\Longrightarrow}$ (rev t) @ [h :: l_2]

$\underset{\text{by def of append}}{\Longrightarrow}$ append (rev t) [h :: l_2] $\underset{\text{by def of rev\_append'}}{\Longrightarrow}$ rev_append' (t, h :: l_2)

$\underset{\text{IH}}{\Longrightarrow}$ rev_append t (h :: l_2)

$\Longrightarrow$ rev-append $l_1\ l_2$ given $l_1 = x :: xs$

def of rev_append

for $x = h$ and $xs = t$

$\therefore$ By Structural Induction,

rev-append' $l_1\ l_2$ = rev-append $l_1\ l_2$

for $l_1 = h :: t$