_A4_

Problem 2: Tail to Head Tales

Theorem: forall lst, rev lst = nrev lst

We need a stronger IH. so, we change the claim as follow:

Theorem. forall lst, forall lst'. irves lst' lst = nrev lst @ lst'

Proof. By induction on lst

P (lst) = forall lst', irev lst' lst = nrev lst @ lst'

```
let rec nrev = function
  | [] -> []
  | h::t -> nrev t @ [h]

let rec irev acc = function
  | [] -> acc
  | h::t -> irev (h::acc) t
let rev lst = irev [] lst
```

Base case: lst = []

show. forall lst', irev lst' [] = nrev [] @ lst'

irev lst' []
= {evaluate irev}
lst'

nrev [] @ lst'
= {evaluate nrev}
[] @ lst'
= {evaluate lst'}
lst'

Inductive case: lst = h::t

show. forall lst', irev lst' (h::t) = nrev (h::t) @ lst'

IH. forall lst', irev lst' t = nrev t @ lst'

irev lst' (h::t)
= {evaluate irev}
irev (h:: lst') t
= {By IH with lst' = h:: lst'}
nrev t @ (h:: lst')

nrev (h::t) @ lst'
= {evaluate nrev}
nrev t @ [h] @ lst'
= {evaluate lst operation}
nrev t @ (h:: lst')

more explanation on these steps
is that since, lst' = h:: lst'
we can get
irev lst' t
and this equals to
nrev t @ lst' by IH.
As lst' = h:: lst
Then nrev t @ (h:: lst')

QED.