```
list_append_singleton_ind  THM

⊢  ∀T:Type, Q:((T List)→Prop).
|  Q(nil)
|  ⇒
|  (∀ys:T List, x:T. Q(ys) ⇒ Q(ys @ [x])) ⇒ (∀zs:T List. Q(zs))
| by Auto THEN Unfold ‘guard‘ 0
|
1. T : Type
2. Q : (T List)→Prop
3. Q(nil)
4. ∀ys:T List, x:T. Q(ys) ⇒ Q(ys @ [x])
⊢  ∀zs:T List. Q(zs)  by Assert ∀n:ℕ, l:T List. ||l|| = n ⇒ Q(l)
|
|\
| .....assertion .....
| 1. T : Type
| 2. Q : (T List)→Prop
| 3. Q(nil)
| 4. ∀ys:T List, x:T. Q(ys) ⇒ Q(ys @ [x])
| ⊢  ∀n:ℕ, l:T List. ||l|| = n ⇒ Q(l)
| | by D 0 THENA Auto THEN NatInd (-1)
| |
| |\
| | .....basecase .....
| | 1. T : Type
| | 2. Q : (T List)→Prop
| | 3. Q(nil)
| | 4. ∀ys:T List, x:T. Q(ys) ⇒ Q(ys @ [x])
| | ⊢  ∀l:T List. ||l|| = 0 ⇒ Q(l)
| |   by Auto’ THEN Subst l = nil
| |      ...
| |
| .\
| . .....upcase .....
| . 1. T : Type
| . 2. Q : (T List)→Prop
| . 3. Q(nil)
| . 4. ∀ys:T List, x:T. Q(ys) ⇒ Q(ys @ [x])
| . 5. n : ℤ
| . 6. 0<n
| . 7. ∀l:T List. ||l|| = n-1 ⇒ Q(l)
| . ⊢  ∀l:T List. ||l|| = n ⇒ Q(l)  by Auto
| . |
| . 1. T : Type
| . 2. Q : (T List)→Prop
| . 3. Q(nil)
| . 4. ∀ys:T List, x:T. Q(ys) ⇒ Q(ys @ [x])
| . 5. n : ℤ
| . 6. 0<n
| . 7. ∀l:T List. ||l|| = n-1 ⇒ Q(l)
| . 8. l : T List
| . 9. ||l|| = n
| . ⊢  Q(l)
| . | by  InstLemma ‘list_decomp_reverse‘ [T;l] THENA Auto’ THEN
```

```
| .
| . |     ...
| . |
| . 1. T : Type
| . 2. Q : (T List)→Prop
| . 3. Q(nil)
| . 4. ∀ys:T List, x:T. Q(ys) ⇒ Q(ys @ [x])
| . 5. n : ℤ
| . 6. 0<n
| . 7. ∀l:T List. ||l|| = n-1 ⇒ Q(l)
| . 8. l : T List
| . 9. x : T
| . 10. L' : T List
| . 11. l = (L' @ [x])
| . ⊢  ||L' @ [x]|| = n ⇒ Q(L' @ [x])
| . | by Subst ||L' @ [x]|| = ||L'||+1 0 THENA Auto
| . |
| . |\
| . | .....equality .....
| . | 1. T : Type
| . | 2. Q : (T List)→Prop
| . | 3. Q(nil)
| . | 4. ∀ys:T List, x:T. Q(ys) ⇒ Q(ys @ [x])
| . | 5. n : ℤ
| . | 6. 0<n
| . | 7. ∀l:T List. ||l|| = n-1 ⇒ Q(l)
| . | 8. l : T List
| . | 9. x : T
| . | 10. L' : T List
| . | 11. l = (L' @ [x])
| . | ⊢  ||L' @ [x]|| = ||L'||+1
| . |   by InstLemma `length_append` [T;L';[x]] THEN Auto
| . |
| . |         THEN Reduce (-1) THEN Auto'
| . |
| . .\
| . . 1. T : Type
| . . 2. Q : (T List)→Prop
| . . 3. Q(nil)
| . . 4. ∀ys:T List, x:T. Q(ys) ⇒ Q(ys @ [x])
| . . 5. n : ℤ
| . . 6. 0<n
| . . 7. ∀l:T List. ||l|| = n-1 ⇒ Q(l)
| . . 8. l : T List
| . . 9. x : T
| . . 10. L' : T List
| . . 11. l = (L' @ [x])

| . . ⊢  ||L'||+1 = n ⇒ Q(L' @ [x])
| . .   by ...
|
.\
. 1. T : Type
. 2. Q : (T List)→Prop
. 3. Q(nil)
. 4. ∀ys:T List, x:T. Q(ys) ⇒ Q(ys @ [x])
```

```
. 5. ∀n:ℕ, l:T List. ||l|| = n ⇒ Q(l)
. ⊢  ∀zs:T List. Q(zs)
.   by Auto THEN AllHyps (\i. InstHyp [||zs||; zs
.      ] i THENA Complete Auto')
. THEN Trivial
END
```