```
append_one_one  THM

⊢  ∀T:Type, x,z,y,w:T List.
|  ||x|| = ||z|| ∨ ||y|| = ||w|| ⇒ (x @ y) = (z @ w) ⇒ x = z & y = w
| by Auto THEN (FwdThru Thm: equal_appends [-1]) THEN ExRepD
|    THEN
|    (Subst (e = nil) -1)
|
|\
| .....equality .....
| 1. T : Type
| 2. x : T List
| 3. z : T List
| 4. y : T List
| 5. w : T List
| 6. ||x|| = ||z|| ∨ ||y|| = ||w||
| 7. (x @ y) = (z @ w)
| 8. e : T List
| 9. x = (z @ e) & w = (e @ y) ∨ z = (x @ e) & y = (e @ w)
| ⊢  e = nil  by Analyze -2
| |
| |\
| | 1. T : Type
| | 2. x : T List
| | 3. z : T List
| | 4. y : T List
| | 5. w : T List
| | 6. ||x|| = ||z|| ∨ ||y|| = ||w||
| | 7. (x @ y) = (z @ w)
| | 8. x = (z @ nil) & w = (nil @ y) ∨ z = (x @ nil) & y = (nil @ w)
| | ⊢  nil = nil  by Obvious
| |
| .\
| . 1. T : Type
| . 2. x : T List
| . 3. z : T List
| . 4. y : T List
| . 5. w : T List
| . 6. ||x|| = ||z|| ∨ ||y|| = ||w||
| . 7. (x @ y) = (z @ w)
| . 8. u : T
| . 9. v : T List
| . 10. x = (z @ [u / v]) & w = ([u / v] @ y)
| .     ∨ z = (x @ [u / v]) & y = ([u / v] @ w)
| . ⊢  [u / v] = nil
| .   by SplitOrHyps
| .      THEN
| .      (AllHyps
| .       (λh.
| .        (ApFunToHypEquands ‘l’ ||l|| ℤ h)
| .        THEN
| .        (RWO
| .         Thm* ∀as,bs:T List. ||as @ bs|| = ||as||+||bs|| ∈ ℤ
| .         {length_append}
| .         -1)))
```

```
|
.\
. 1. T : Type
. 2. x : T List
. 3. z : T List
. 4. y : T List
. 5. w : T List
. 6. ||x|| = ||z|| ∨ ||y|| = ||w||
. 7. (x @ y) = (z @ w)
. 8. T List
. 9. x = (z @ nil) & w = (nil @ y) ∨ z = (x @ nil) & y = (nil @ w)
. ⊢  x = z & y = w
.   by ((Reduce -1)
.       THEN
.       (RWO Thm* ∀l:T List. (l @ nil) ~ l    {append_nil_sq} -1))
.      THEN
.      (Analyze -1)
END
```