```
append_is_nil  THM

⊢  ∀T:Type, l1,l2:T List. (l1 @ l2) = nil ⟺ l1 = nil & l2 = nil
| by Assert ∀T:Type, x:T, l:T List. [x / l] = nil ⇒ False
|
|\
| .....assertion .....
| ⊢  ∀T:Type, x:T, l:T List. [x / l] = nil ⇒ False
| | by Auto THEN
| ApFunToHypEquands 'z' ||z|| ℕ (-1)
| THENA Auto'
| |
| 1. T : Type
| 2. x : T
| 3. l : T List
| 4. [x / l] = nil
| 5. ||[x / l]|| = ||nil||
| ⊢  False  by Reduce (-1)
| |
| 1. T : Type
| 2. x : T
| 3. l : T List
| 4. [x / l] = nil
| 5. ||l||+1 = 0
| ⊢  False  by Auto
|
.\
. 1. ∀T:Type, x:T, l:T List. [x / l] = nil ⇒ False
. ⊢  ∀T:Type, l1,l2:T List. (l1 @ l2) = nil ⟺ l1 = nil & l2 = nil
. | by D 0 THENM D 0 THENA Auto THEN ListInd (-1)
.
. |    THEN Reduce 0 THEN Try (Complete Auto)
. |
. 1. ∀T:Type, x:T, l:T List. [x / l] = nil ⇒ False
. 2. T : Type
. 3. T List
. 4. u : T
. 5. v : T List
. 6. ∀l2:T List. (v @ l2) = nil ⟺ v = nil & l2 = nil
. ⊢  ∀l2:T List. [u / (v @ l2)] = nil ⟺ [u / v] = nil & l2 = nil
.   by D 0 THENA Auto THEN ListInd (-1) THEN Auto
.
.     ...
.
END
```