```
is_member_non_nil  THM

⊢  ∀eq:{T=₂}, u:T, L:T List. u(∈eq) L ⇒ (∃v:T, M:T List. L = v.M)
| by Analyze 0 THEN Analyze 0 THEN DiscreteEq -1 THEN UnivCD
|
1. T : Type
2. eq : {T=₂}
3. eq ∈ T→T→𝔹
4. ∀x,y:T. eq(x,y) ⟺ x = y
5. u : T
6. L : T List
7. u(∈eq) L
⊢  ∃v:T, M:T List. L = v.M
| by Inst
|    Thm* ∀eq:{T=₂}, L:T List, x:T.
|         x(∈eq) L ⇒ (∃M,N:T List. L = (M @ (x.N)))
|    {is_member_append_lemma}
|    [T;eq;L;u]
|
8. ∃M,N:T List. L = (M @ (u.N))
⊢  ∃v:T, M:T List. L = v.M  by Repeat (Analyze -1)
|
8. M : T List
9. N : T List
10. L = (M @ (u.N))
⊢  ∃v:T, M:T List. L = v.M  by ListInd 8
|
|\
| 9. N : T List
| ⊢  L = (nil @ (u.N)) ⇒ (∃v:T, M:T List. L = v.M)
| | by Reduce 0 THEN Analyze 0
| |
| 10. L = u.N
| ⊢  ∃v:T, M:T List. L = v.M  by Witness u THEN Witness N
|
.\
. 10. u1 : T
. 11. v : T List
. 12. L = (v @ (u.N)) ⇒ (∃v:T, M:T List. L = v.M)
. ⊢  L = ((u1.v) @ (u.N)) ⇒ (∃v:T, M:T List. L = v.M)
. | by Reduce 0 THEN Analyze 0
. |
. 13. L = u1.(v @ (u.N))
. ⊢  ∃v:T, M:T List. L = v.M  by Witness u1 THEN Witness v @ (u.N)
END
```