```
sublist_list_all_lemma  THM

⊢  ∀EQ:{T=₂}, eq:{T≡₂}, L,M:T List.
|  L(⊆eq)M ⟺ (∀z:T. z(∈EQ) L ⇒ z(∈eq) M)  by UnivCD
|
1. T : Type
2. EQ : {T=₂}
3. eq : {T≡₂}
4. L : T List
5. M : T List
⊢  L(⊆eq)M ⟺ (∀z:T. z(∈EQ) L ⇒ z(∈eq) M)
| by Equivalence 3 THEN DiscreteEq 2 THEN Analyze 0 THEN Analyze 0
|
|\
| 3. EQ ∈ T→T→𝔹
| 4. ∀x,y:T. EQ(x,y) ⟺ x = y
| 5. eq : {T≡₂}
| 6. eq ∈ T→T→𝔹
| 7. ∀x:T. eq(x,x)
| 8. ∀x,y:T. eq(x,y) ⟺ eq(y,x)
| 9. ∀x,y,z:T. eq(x,y) ⇒ eq(y,z) ⇒ eq(x,z)
| 10. L : T List
| 11. M : T List
| 12. L(⊆eq)M
| ⊢  ∀z:T. z(∈EQ) L ⇒ z(∈eq) M
| | by UnfoldTopAb 12
| |    THEN
| |    Inst
| |    Thm* ∀eq:{T=₂}, P:(T→Prop), L:T List.
| |         ∀x∈L.P(x) ⟺ (∀z:T. z(∈eq) L ⇒ P(z))
| |    {list_all_is_member_lemma}
| |    [T;EQ;λ₂x. x(∈eq) M;L]
| |
| 12. ∀x∈L.x(∈eq) M
| 13. ∀x∈L.x(∈eq) M ⟺ (∀z:T. z(∈EQ) L ⇒ z(∈eq) M)
| ⊢  ∀z:T. z(∈EQ) L ⇒ z(∈eq) M  by Analyze -1 THEN Thin -1
| |
| 13. ∀x∈L.x(∈eq) M ⇒ (∀z:T. z(∈EQ) L ⇒ z(∈eq) M)
| ⊢  ∀z:T. z(∈EQ) L ⇒ z(∈eq) M
|   by RenameVar `xx` 12 THEN With xx (Analyze -1) THEN Trivial
|
.\
. 3. EQ ∈ T→T→𝔹
. 4. ∀x,y:T. EQ(x,y) ⟺ x = y
. 5. eq : {T≡₂}
. 6. eq ∈ T→T→𝔹
. 7. ∀x:T. eq(x,x)
. 8. ∀x,y:T. eq(x,y) ⟺ eq(y,x)
. 9. ∀x,y,z:T. eq(x,y) ⇒ eq(y,z) ⇒ eq(x,z)
. 10. L : T List
. 11. M : T List
. 12. ∀z:T. z(∈EQ) L ⇒ z(∈eq) M
. ⊢  L(⊆eq)M
. | by UnfoldTopAb 0
. |    THEN
```

```
. |    Inst
. |    Thm* ∀eq:{T=₂}, P:(T→Prop), L:T List.
. |         ∀x∈L.P(x) ⟺ (∀z:T. z(∈eq) L ⇒ P(z))
. |    {list_all_is_member_lemma}
. |    [T;EQ;λ₂x. x(∈eq) M;L]
. |
. 13. ∀x∈L.x(∈eq) M ⟺ (∀z:T. z(∈EQ) L ⇒ z(∈eq) M)
. ⊢  ∀x∈L.x(∈eq) M  by Analyze -1 THEN Thin -2
. |
. 13. ∀x∈L.x(∈eq) M ⇐ (∀z:T. z(∈EQ) L ⇒ z(∈eq) M)
. ⊢  ∀x∈L.x(∈eq) M
.   by RenameVar ‘xx’ 12 THEN With xx (Analyze -1) THEN Trivial
END
```