```
sublist_iff_assert_sublist_2_x  THM

⊢  ∀L,M:T List, eq:{T=₂}. L(⊆eq)M ⟺ (⊆₂eq)(L,M)  by UnivCD
|
1. T : Type
2. L : T List
3. M : T List
4. eq : {T=₂}
⊢  L(⊆eq)M ⟺ (⊆₂eq)(L,M)
| by Analyze 2
|    THEN
|    Rewrite
|    (HigherC sublist_unrollC THENC HigherC sublist_2_unrollC
|     THENC
|     TryC (HigherC assert_evalC))
|    0
|
|\
| ⊢  True ⟺ True  by Auto
|
.\
. 5. u : T
. 6. v : T List
. ⊢  u(∈eq) M & v(⊆eq)M ⟺ (u(∈eq) M∧₂∀x∈₂v.x(∈eq) M)
. | by DiscreteEq 4
. |    THEN
. |    Inst Thm* ∀p,q:𝔹. (p∧₂q) ⟺ p & q    {assert_of_band}
. |    [u(∈eq) M;∀x∈₂v.x(∈eq) M]
. |
. 5. eq ∈ T→T→𝔹
. 6. ∀x,y:T. eq(x,y) ⟺ x = y
. 7. u : T
. 8. v : T List
. 9. (u(∈eq) M∧₂∀x∈₂v.x(∈eq) M) ⟺ u(∈eq) M & ∀x∈₂v.x(∈eq) M
. ⊢  u(∈eq) M & v(⊆eq)M ⟺ (u(∈eq) M∧₂∀x∈₂v.x(∈eq) M)
. | by Analyze 0 THEN Analyze 0
. |
. |\
. | 10. u(∈eq) M & v(⊆eq)M
. | ⊢  u(∈eq) M∧₂∀x∈₂v.x(∈eq) M  by HypBackchain
. | |
. | ⊢  ∀x∈₂v.x(∈eq) M
. | | by BackThru
. | |    Thm* ∀L:T List, P:(T→𝔹). ∀x∈L.P(x) ⟺ ∀x∈₂L.P(x)
. | |    {list_all_iff_assert_list_all_2}
. | |
. | ⊢  ∀x∈v.x(∈eq) M  by Fold ‘sublist‘ 0
. | |
. | ⊢  v(⊆eq)M  by Auto
. |
. .\
. . 10. u(∈eq) M∧₂∀x∈₂v.x(∈eq) M
. . ⊢  u(∈eq) M & v(⊆eq)M
. . | by RWH (LemmaC Thm* ∀p,q:𝔹. (p∧₂q) ⟺ p & q    {assert_of_band})
. . |    -1
```

```
. . |    THEN
. . |    OnHyps [-1;0] Analyze
. . |
. . |\
. . | 10. u(∈eq) M
. . | 11. ∀x∈₂v.x(∈eq) M
. . | ⊢  u(∈eq) M  by Trivial
. . |
. . .\
. . . 10. u(∈eq) M
. . . 11. ∀x∈₂v.x(∈eq) M
. . . ⊢  v(⊆eq)M  by UnfoldTopAb 0
. . . |
. . . ⊢  ∀x∈v.x(∈eq) M
. . .   by BackThru
. . .      Thm* ∀L:T List, P:(T→𝔹). ∀x∈L.P(x) ⟺ ∀x∈₂L.P(x)
. . .      {list_all_iff_assert_list_all_2}
END
```