```
list_iso_iff_assert_list_iso_2  THM

⊢  Discrete{T} ⇒ (∀eq:{T≡₂}, L,M:T List. L(∼eq)M ⟺ (∼₂eq)(L,M))
| by UnivCD
|
1. T : Type
2. Discrete{T}
3. eq : {T≡₂}
4. L : T List
5. M : T List
⊢  L(∼eq)M ⟺ (∼₂eq)(L,M)
| by Unfold ‘list_iso‘ 0 THEN Unfold ‘list_iso_2‘ 0 THEN Reduce 0
|
⊢  L(⊆eq)M & M(⊆eq)L ⟺ (((⊆₂eq)(L,M))∧₂((⊆₂eq)(M,L)))
| by Equivalence 3
|    THEN
|    Rewrite
|    (HigherC
|     (LemmaC Thm* ∀p,q:𝔹. (p∧₂q) ⟺ p & q    {assert_of_band}))
|    0
|
4. eq ∈ T→T→𝔹
5. ∀x:T. eq(x,x)
6. ∀x,y:T. eq(x,y) ⟺ eq(y,x)
7. ∀x,y,z:T. eq(x,y) ⇒ eq(y,z) ⇒ eq(x,z)
8. L : T List
9. M : T List
⊢  L(⊆eq)M & M(⊆eq)L ⟺ (⊆₂eq)(L,M) & (⊆₂eq)(M,L)
| by Analyze 0 THEN Analyze 0
|
|\
| 10. L(⊆eq)M & M(⊆eq)L
| ⊢  (⊆₂eq)(L,M) & (⊆₂eq)(M,L)  by Analyze -1 THEN Analyze 0
| |
| |\
| | 10. L(⊆eq)M
| | 11. M(⊆eq)L
| | ⊢  (⊆₂eq)(L,M)
| |   by BackThru
| |      Thm* Discrete{T}
| |           ⇒
| |           (∀eq:{T≡₂}, L,M:T List. L(⊆eq)M ⟺ (⊆₂eq)(L,M))
| |      {sublist_iff_assert_sublist_2}
| |
| .\
| . 10. L(⊆eq)M
| . 11. M(⊆eq)L
| . ⊢  (⊆₂eq)(M,L)
| .   by BackThru
| .      Thm* Discrete{T}
| .           ⇒
| .           (∀eq:{T≡₂}, L,M:T List. L(⊆eq)M ⟺ (⊆₂eq)(L,M))
| .      {sublist_iff_assert_sublist_2}
|
.\
```

```
. 10. (⊆₂eq)(L,M) & (⊆₂eq)(M,L)
. ⊢  L(⊆eq)M & M(⊆eq)L  by Analyze -1 THEN Analyze 0
. |
. |\
. | 10. (⊆₂eq)(L,M)
. | 11. (⊆₂eq)(M,L)
. | ⊢  L(⊆eq)M
. |   by BackThru
. |      Thm* Discrete{T}
. |           ⇒
. |           (∀eq:{T≡₂}, L,M:T List. L(⊆eq)M ⟺ (⊆₂eq)(L,M))
. |      {sublist_iff_assert_sublist_2}
. |
. .\
. . 10. (⊆₂eq)(L,M)
. . 11. (⊆₂eq)(M,L)
. . ⊢  M(⊆eq)L
. .   by BackThru
. .      Thm* Discrete{T}
. .           ⇒
. .           (∀eq:{T≡₂}, L,M:T List. L(⊆eq)M ⟺ (⊆₂eq)(L,M))
. .      {sublist_iff_assert_sublist_2}
END
```