append_functionality_wrt_list_iso THM

```
⊢  ∀L1,L2:T List, eq1,eq2,eq3:{T≡₂}, L3,L4:T List.
|  Discrete{T}
|  ⇒
|  eq1 = eq2
|  ⇒
|  eq2 = eq3 ⇒ L1(∼eq1)L2 ⇒ L3(∼eq2)L4 ⇒ (L1 @ L3)(∼eq3)(L2 @ L4)
| by RepeatFor 3 (Analyze 0)
|    THEN
|    RepeatFor 3 (Analyze 0 THEN Equivalence -1)
|    THEN
|    UnivCD
|
1. T : Type
2. L1 : T List
3. L2 : T List
4. eq1 : {T≡₂}
5. eq1 ∈ T→T→𝔹
6. ∀x:T. eq1(x,x)
7. ∀x,y:T. eq1(x,y) ⟺ eq1(y,x)
8. ∀x,y,z:T. eq1(x,y) ⇒ eq1(y,z) ⇒ eq1(x,z)
9. eq2 : {T≡₂}
10. eq2 ∈ T→T→𝔹
11. ∀x:T. eq2(x,x)
12. ∀x,y:T. eq2(x,y) ⟺ eq2(y,x)
13. ∀x,y,z:T. eq2(x,y) ⇒ eq2(y,z) ⇒ eq2(x,z)
14. eq3 : {T≡₂}
15. eq3 ∈ T→T→𝔹
16. ∀x:T. eq3(x,x)
17. ∀x,y:T. eq3(x,y) ⟺ eq3(y,x)
18. ∀x,y,z:T. eq3(x,y) ⇒ eq3(y,z) ⇒ eq3(x,z)
19. L3 : T List
20. L4 : T List
21. Discrete{T}
22. eq1 = eq2
23. eq2 = eq3
24. L1(∼eq1)L2
25. L3(∼eq2)L4
⊢  (L1 @ L3)(∼eq3)(L2 @ L4)
| by HypSubst 22 24 THENA Equivalence -1 THEN HypSubst 23 24
|    THENA
|    Equivalence -1
|    THEN
|    HypSubst 23 25
|    THENA
|    Equivalence -1
|    THEN
|    RenameVar ‘eq’ 14
|    THEN
|    OnHyps [23;22;13;12;11;10;9;8;7;6;5;4] (λi.Thin i)
|
4. eq : {T≡₂}
5. eq ∈ T→T→𝔹
6. ∀x:T. eq(x,x)
```

```
7. ∀x,y:T. eq(x,y) ⟺ eq(y,x)
8. ∀x,y,z:T. eq(x,y) ⇒ eq(y,z) ⇒ eq(x,z)
9. L3 : T List
10. L4 : T List
11. Discrete{T}
12. L1(∼eq)L2
13. L3(∼eq)L4
⊢  (L1 @ L3)(∼eq)(L2 @ L4)
| by OnHyps [13;12;0] (λi.UnfoldTopAb i THEN Analyze i)
|
|\
| 12. L1(⊆eq)L2
| 13. L2(⊆eq)L1
| 14. L3(⊆eq)L4
| 15. L4(⊆eq)L3
| ⊢  (L1 @ L3)(⊆eq)(L2 @ L4)
|   by Using [‘eq1’,eq;‘eq2’,eq]
|      (BackThru
|       Thm* ∀L1,L2:T List, eq1,eq2,eq3:{T≡₂}, L3,L4:T List.
|            Discrete{T}
|            ⇒
|            eq1 = eq2
|            ⇒
|            eq2 = eq3
|            ⇒
|            L1(⊆eq1)L2 ⇒ L3(⊆eq2)L4 ⇒ (L1 @ L3)(⊆eq3)(L2 @ L4)
|       {append_functionality_wrt_sublist})
|
.\
. 12. L1(⊆eq)L2
. 13. L2(⊆eq)L1
. 14. L3(⊆eq)L4
. 15. L4(⊆eq)L3
. ⊢  (L2 @ L4)(⊆eq)(L1 @ L3)
.   by Using [‘eq1’,eq;‘eq2’,eq]
.      (BackThru
.       Thm* ∀L1,L2:T List, eq1,eq2,eq3:{T≡₂}, L3,L4:T List.
.            Discrete{T}
.            ⇒
.            eq1 = eq2
.            ⇒
.            eq2 = eq3
.            ⇒
.            L1(⊆eq1)L2 ⇒ L3(⊆eq2)L4 ⇒ (L1 @ L3)(⊆eq3)(L2 @ L4)
.       {append_functionality_wrt_sublist})
END
```