```
append_commutes_under_list_iso  THM

⊢  Discrete{T} ⇒ (∀eq:{T≡₂}, L,M:T List. (L @ M)(∼eq)(M @ L))
| by Unfold ‘list_iso‘ 0 THEN UnivCD THEN Analyze 0
|
|\
| 1. T : Type
| 2. Discrete{T}
| 3. eq : {T≡₂}
| 4. L : T List
| 5. M : T List
| ⊢  (L @ M)(⊆eq)(M @ L)
|   by BackThru
|      Thm* Discrete{T}
|           ⇒
|           (∀eq:{T≡₂}, L,M:T List. (L @ M)(⊆eq)(M @ L))
|      {append_commutes_under_sublist}
|
.\
. 1. T : Type
. 2. Discrete{T}
. 3. eq : {T≡₂}
. 4. L : T List
. 5. M : T List
. ⊢  (M @ L)(⊆eq)(L @ M)
.   by BackThru
.      Thm* Discrete{T}
.           ⇒
.           (∀eq:{T≡₂}, L,M:T List. (L @ M)(⊆eq)(M @ L))
.      {append_commutes_under_sublist}
END
```