```
sublist_weakening_wrt_identity  THM

⊢  Discrete{T} ⇒ (∀eq:{T≡₂}, L,M:T List. L = M ⇒ L(⊆eq)M)
| by UnivCD
|
1. T : Type
2. Discrete{T}
3. eq : {T≡₂}
4. L : T List
5. M : T List
6. L = M
⊢  L(⊆eq)M  by RevHypSubst 6 0 THENA Equivalence 3
|
⊢  L(⊆eq)L
| by Analyze 4 THEN Rewrite (HigherC sublist_unrollC) 0
|    THEN
|    Try Trivial
|
7. u : T
8. v : T List
⊢  u(∈eq) (u.v) & v(⊆eq)(u.v)  by Analyze 0
|
|\
| ⊢  u(∈eq) (u.v)  by Rewrite (HigherC is_member_unrollC) 0
| |
| ⊢  if eq(u,u)→ true₂ else u(∈eq) v fi  by Equivalence 3
| |
| 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
| 10. L = M
| 11. u : T
| 12. v : T List
| ⊢  if eq(u,u)→ true₂ else u(∈eq) v fi
| | by Witness5 u
| |    THEN
| |    FwdThru Thm* ∀P:𝔹. P ⟺ P = true₂    {assert_iff_btrue} [-1]
| |
| 5. ∀x,y:T. eq(x,y) ⟺ eq(y,x)
| 6. ∀x,y,z:T. eq(x,y) ⇒ eq(y,z) ⇒ eq(x,z)
| 7. L : T List
| 8. M : T List
| 9. L = M
| 10. u : T
| 11. v : T List
| 12. eq(u,u)
| 13. eq(u,u) = true₂
| ⊢  if eq(u,u)→ true₂ else u(∈eq) v fi
|   by HypSubst -1 0
|      THEN
|      Rewrite (HigherC ifthenelse_evalC THENC assert_evalC) 0
|      THEN
```

```
|      Trivial
|
.\
. ⊢  v(⊆eq)(u.v)
.   by BackThru
.      Thm* Discrete{T} ⇒ (∀eq:{T≡₂}, u:T, v:T List. v(⊆eq)(u.v))
.      {sublist_tail}
END
```