```
sublist_tail  THM

⊢  Discrete{T} ⇒ (∀eq:{T≡₂}, u:T, v:T List. v(⊆eq)(u.v))  by UnivCD
|
1. T : Type
2. Discrete{T}
3. eq : {T≡₂}
4. u : T
5. v : T List
⊢  v(⊆eq)(u.v)
| by Rewrite (UnfoldC ‘sublist‘ THENC HigherC is_member_unrollC) 0
|
⊢  ∀x∈v.if eq(x,u)→ true₂ else x(∈eq) v fi
| by FwdThru
|    Thm* Discrete{T} ⇒ (∃f:{T=₂}. True)
|    {discrete_implies_discrete_equality}
|    [2]
|    THEN
|    RenameVar ‘f’ -1
|
6. ∃f:{T=₂}. True
⊢  ∀x∈v.if eq(x,u)→ true₂ else x(∈eq) v fi  by Analyze -1
|
6. {T=₂}
7. True
⊢  ∀x∈v.if eq(x,u)→ true₂ else x(∈eq) v fi
| by Inst
|    Thm* ∀eq:{T=₂}, f:{T≡₂}, x,y:T. eq(x,y) ⇒ f(x,y)
|    {equivalence_weakening_lemma}
|    [T;f1;eq]
|
8. ∀x,y:T. f1(x,y) ⇒ eq(x,y)
⊢  ∀x∈v.if eq(x,u)→ true₂ else x(∈eq) v fi  by EquivT 3 THEN Thin -2
|
4. eq ∈ T→T→𝔹
5. u : T
6. v : T List
7. f1 : {T=₂}
8. ∀x,y:T. f1(x,y) ⇒ eq(x,y)
⊢  ∀x∈v.if eq(x,u)→ true₂ else x(∈eq) v fi
| by Inst
|    Thm* ∀eq:{T=₂}, P:(T→Prop), L:T List.
|         ∀x∈L.P(x) ⟺ (∀z:T. z(∈eq) L ⇒ P(z))
|    {list_all_is_member_lemma}
|    [T;f1;λx.if eq(x,u)→ true₂ else x(∈eq) v fi;v]
|    THEN
|    ReduceSOAps -1
|
9. ∀x∈v.if eq(x,u)→ true₂ else x(∈eq) v fi
|  ⟺
|  (∀z:T. z(∈f1) v ⇒ if eq(z,u)→ true₂ else z(∈eq) v fi)
⊢  ∀x∈v.if eq(x,u)→ true₂ else x(∈eq) v fi  by HypBackchain
|
10. z : T
11. z(∈f1) v
```

```
⊢  if eq(z,u)→ true₂ else z(∈eq) v fi  by IfThenElseCases 0
|
|\
| .....truecase .....
| 12. eq(z,u) = true₂
| ⊢  true₂  by Rewrite (HigherC assert_evalC) 0 THEN Trivial
|
.\
. .....falsecase .....
. 12. eq(z,u) = false₂
. ⊢  z(∈eq) v
. | by Using [‘eq’,f1]
. |    (BackThru
. |     Thm* ∀eq:{T=₂}, u:T, L:T List. u(∈eq) L ⇒ (∀f:{T≡₂}. u(∈f) L)
. |     {is_member_equalities_lemma})
. |
. ⊢  z(∈f1) v  by Trivial
END
```