```
sublist_tail1  THM

⊢  ∀eq:{T=₂}, u:T, v:T List. v(⊆eq)(u.v)  by UnivCD
|
1. T : Type
2. eq : {T=₂}
3. u : T
4. v : T List
⊢  v(⊆eq)(u.v)  by Unfold ‘sublist‘ 0
|
⊢  ∀x∈v.x(∈eq) (u.v)  by Rewrite (HigherC is_member_unrollC) 0
|
⊢  ∀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;eq;λz.if eq(z,u)→ true₂ else z(∈eq) v fi;v]
|    THENL
|    [Auto;Auto;Analyze 2;Auto;Id]
|
5. ∀x∈v.(λz.if eq(z,u)→ true₂ else z(∈eq) v fi)(x)
|  ⟺
|  (∀z:T. z(∈eq) v ⇒ (λz.if eq(z,u)→ true₂ else z(∈eq) v fi)(z))
⊢  ∀x∈v.if eq(x,u)→ true₂ else x(∈eq) v fi
| by ReduceSOAps 5 THEN Analyze 5 THEN Thin -2 THEN BackThru -1
|
5. ∀x∈v.if eq(x,u)→ true₂ else x(∈eq) v fi ⇐ (∀z:T.
|  z(∈eq) v ⇒ if eq(z,u)→ true₂ else z(∈eq) v fi)
⊢  ∀z:T. z(∈eq) v ⇒ if eq(z,u)→ true₂ else z(∈eq) v fi
| by Analyze 2 THEN UnivCD
|
2. eq : T→T→ℙ
3. (∀x,y:T. eq(x,y) ⟺ x = y) & eq ∈ T→T→ℙ
4. u : T
5. v : T List
6. ∀x∈v.if eq(x,u)→ true₂ else x(∈eq) v fi ⇐ (∀z:T.
|  z(∈eq) v ⇒ if eq(z,u)→ true₂ else z(∈eq) v fi)
7. z : T
8. z(∈eq) v
⊢  if eq(z,u)→ true₂ else z(∈eq) v fi
| by Rewrite (LemmaC Thm* ∀P:ℙ. P ⟺ P = true₂    {assert_iff_btrue})
|    8
|
8. z(∈eq) v = true₂
⊢  if eq(z,u)→ true₂ else z(∈eq) v fi  by HypSubst 8 0
|
⊢  if eq(z,u)→ true₂ else true₂ fi
  by IfThenElseCases 0 THEN Rewrite (HigherC assert_evalC) 0
     THEN
     Trivial
END
```