```
disjoint_iff_assert_disjoint2  THM

⊢  ∀eq:{T≡₂}, L,M:T List. disjoint(eq;L;M) ⟺ disjoint₂(eq;L;M)
| by UnivCD THEN EquivT 2
|
1. T : Type
2. eq : {T≡₂}
3. eq ∈ T→T→𝔹
4. L : T List
5. M : T List
⊢  disjoint(eq;L;M) ⟺ disjoint₂(eq;L;M)
| by ListInd 4
|    THEN
|    Rewrite
|    (HigherC disjoint2_unrollC THENC HigherC disjoint_unrollC)
|    0
|
|\
| ⊢  True ⟺ true₂  by Rewrite (HigherC assert_evalC) 0
|
.\
. 6. u : T
. 7. v : T List
. 8. disjoint(eq;v;M) ⟺ disjoint₂(eq;v;M)
. ⊢  ¬₂u(∈eq) M & disjoint(eq;v;M)
. |  ⟺
. |  if u(∈eq) M→ false₂ else disjoint₂(eq;v;M) fi
. | by IfThenElseCases 0 THEN BoolEval
. |
. |\
. | 9. u(∈eq) M = true₂
. | ⊢  false₂ & disjoint(eq;v;M) ⟺ false₂
. |   by Rewrite (HigherC assert_evalC) 0
. |
. .\
. . 9. u(∈eq) M = false₂
. . ⊢  true₂ & disjoint(eq;v;M) ⟺ disjoint₂(eq;v;M)
. . | by Rewrite (HigherC assert_evalC) 0
. . |
. . |\
. . | 8. disjoint(eq;v;M) ⇒ disjoint₂(eq;v;M)
. . | 9. disjoint(eq;v;M) ⇐ disjoint₂(eq;v;M)
. . | 10. u(∈eq) M = false₂
. . | 11. True
. . | 12. disjoint(eq;v;M)
. . | ⊢  disjoint₂(eq;v;M)  by HypBackchain THEN Trivial
. . |
. . .\
. . . 8. disjoint(eq;v;M) ⇒ disjoint₂(eq;v;M)
. . . 9. disjoint(eq;v;M) ⇐ disjoint₂(eq;v;M)
. . . 10. u(∈eq) M = false₂
. . . 11. disjoint₂(eq;v;M)
. . . ⊢  disjoint(eq;v;M)  by HypBackchain THEN Trivial
END
```