```
append_commutes_under_is_member  THM

⊢  ∀eq:{T≡₂}, L,M:T List, x:T. x(∈eq) (L @ M) ⟺ x(∈eq) (M @ L)
| by UnivCD THEN Equivalence 2 THEN Repeat (Analyze 0)
|
|\
| 1. T : Type
| 2. eq : {T≡₂}
| 3. eq ∈ T→T→𝔹
| 4. ∀x:T. eq(x,x)
| 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. x : T
| 10. x(∈eq) (L @ M)
| ⊢  x(∈eq) (M @ L)
| | by FwdThru
| |    Thm* ∀eq:{T≡₂}, L,M:T List, x:T.
| |         x(∈eq) (L @ M) ⟺ x(∈eq) L ∨ x(∈eq) M
| |    {is_member_of_append_lemma}
| |    [-1]
| |
| 11. x(∈eq) L ∨ x(∈eq) M
| ⊢  x(∈eq) (M @ L)
| | by BackThru
| |    Thm* ∀eq:{T≡₂}, L,M:T List, x:T.
| |         x(∈eq) (L @ M) ⟺ x(∈eq) L ∨ x(∈eq) M
| |    {is_member_of_append_lemma}
| |
| ⊢  x(∈eq) M ∨ x(∈eq) L  by Analyze -1
| |
| |\
| | 11. x(∈eq) L
| | ⊢  x(∈eq) M ∨ x(∈eq) L  by Choose [2]
| |
| .\
| . 11. x(∈eq) M
| . ⊢  x(∈eq) M ∨ x(∈eq) L  by Choose [1]
|
.\
. 1. T : Type
. 2. eq : {T≡₂}
. 3. eq ∈ T→T→𝔹
. 4. ∀x:T. eq(x,x)
. 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. x : T
. 10. x(∈eq) (M @ L)
. ⊢  x(∈eq) (L @ M)
. | by FwdThru
. |    Thm* ∀eq:{T≡₂}, L,M:T List, x:T.
. |         x(∈eq) (L @ M) ⟺ x(∈eq) L ∨ x(∈eq) M
```

```
. |    {is_member_of_append_lemma}
. |    [-1]
. |
. 11. x(∈eq) M ∨ x(∈eq) L
. ⊢  x(∈eq) (L @ M)
. | by BackThru
. |    Thm* ∀eq:{T≡₂}, L,M:T List, x:T.
. |         x(∈eq) (L @ M) ⟺ x(∈eq) L ∨ x(∈eq) M
. |    {is_member_of_append_lemma}
. |
. ⊢  x(∈eq) L ∨ x(∈eq) M  by Analyze -1
. |
. |\
. | 11. x(∈eq) M
. | ⊢  x(∈eq) L ∨ x(∈eq) M  by Choose [2]
. |
. .\
. . 11. x(∈eq) L
. . ⊢  x(∈eq) L ∨ x(∈eq) M  by Choose [1]
END
```