```
list_exists_is_member_append_lemma  THM

⊢  ∀P:(T→Prop), L:T List.
|  ∃x∈L.P(x) ⟺ (∃M,N:T List, x:T. P(x) & L = (M @ (x.N)))  by UnivCD
|
1. T : Type
2. P : T→Prop
3. L : T List
⊢  ∃x∈L.P(x) ⟺ (∃M,N:T List, x:T. P(x) & L = (M @ (x.N)))
| by Repeat (Analyze 0)
|
|\
| 4. ∃x∈L.P(x)
| ⊢  ∃M,N:T List, x:T. P(x) & L = (M @ (x.N))  by MoveToConcl -1
| |
| 3. L : T List
| ⊢  ∃x∈L.P(x) ⇒ (∃M,N:T List, x:T. P(x) & L = (M @ (x.N)))
| | by ListInd -1
| |
| |\
| | ⊢  ∃x∈nil.P(x) ⇒ (∃M,N:T List, x:T. P(x) & nil = (M @ (x.N)))
| |   by AbReduce 0
| |
| .\
| . 4. u : T
| . 5. v : T List
| . 6. ∃x∈v.P(x) ⇒ (∃M,N:T List, x:T. P(x) & v = (M @ (x.N)))
| . ⊢  ∃x∈(u.v).P(x) ⇒ (∃M,N:T List, x:T. P(x) & u.v = (M @ (x.N)))
| . | by Analyze 0
| . |
| . 7. ∃x∈(u.v).P(x)
| . ⊢  ∃M,N:T List, x:T. P(x) & u.v = (M @ (x.N))
| . | by AbReduce -1 THEN Analyze -1
| . |
| . |\
| . | 7. P(u)
| . | ⊢  ∃M,N:T List, x:T. P(x) & u.v = (M @ (x.N))  by Witness nil
| . | |
| . | ⊢  ∃N:T List, x:T. P(x) & u.v = (nil @ (x.N))  by Witness v
| . | |
| . | ⊢  ∃x:T. P(x) & u.v = (nil @ (x.v))  by AbReduce 0
| . | |
| . | ⊢  ∃x:T. P(x) & u.v = x.v  by Witness u
| . |
| . .\
| . . 7. ∃x∈v.P(x)
| . . ⊢  ∃M,N:T List, x:T. P(x) & u.v = (M @ (x.N))
| . . | by FwdThru -2 [-1] THEN Repeat (Analyze -1)
| . . |
| . . 8. M : T List
| . . 9. N : T List
| . . 10. x : T
| . . 11. P(x)
| . . 12. v = (M @ (x.N))
| . . ⊢  ∃M,N:T List, x:T. P(x) & u.v = (M @ (x.N))  by HypSubst -1 0
```

```
| . . |
| . . ⊢  ∃M@0,N@0:T List, x@0:T.
| . . |  P(x@0) & u.(M @ (x.N)) = (M@0 @ (x@0.N@0))  by Witness u.M
| . . |
| . . ⊢  ∃N@0:T List, x@0:T.
| . . |  P(x@0) & u.(M @ (x.N)) = ((u.M) @ (x@0.N@0))  by Witness N
| . . |
| . . ⊢  ∃x@0:T. P(x@0) & u.(M @ (x.N)) = ((u.M) @ (x@0.N))
| . . | by Witness x
| . . |
| . . ⊢  P(x) & u.(M @ (x.N)) = ((u.M) @ (x.N))  by AbReduce 0
|
.\
. 4. ∃M,N:T List, x:T. P(x) & L = (M @ (x.N))
. ⊢  ∃x∈L.P(x)  by Repeat (Analyze -1)
. |
. 4. M : T List
. 5. N : T List
. 6. x : T
. 7. P(x)
. 8. L = (M @ (x.N))
. ⊢  ∃x∈L.P(x)  by HypSubst -1 0
. |
. ⊢  ∃x∈(M @ (x.N)).P(x)
. | by BackThru
. |    Thm* ∀P:(T→Prop), L,M:T List.
. |         ∃x∈(L @ M).P(x) ⟺ ∃x∈L.P(x) ∨ ∃x∈M.P(x)
. |    {list_exists_append_lemma}
. |
. ⊢  ∃x∈M.P(x) ∨ ∃x∈(x.N).P(x)  by AbReduce 0
. |
. ⊢  ∃x∈M.P(x) ∨ P(x) ∨ ∃x∈N.P(x)  by Choose [2;1]
END
```