```
append_is_singleton  THM

⊢  ∀T:Type, a,b:T List, x:T.
|  [x] = (a @ b) ⇒ a = nil & b = [x] ∨ b = nil & a = [x]
| by InductionOnList THEN (Reduce 0)
|
|\
| 1. T : Type
| 2. T List
| ⊢  ∀b:T List, x:T.
|    [x] = b ⇒ nil = nil & b = [x] ∨ b = nil & nil = [x]  by Obvious
|
.\
. 1. T : Type
. 2. T List
. 3. u : T
. 4. v : T List
. 5. ∀b:T List, x:T.
. |  [x] = (v @ b) ⇒ v = nil & b = [x] ∨ b = nil & v = [x]
. ⊢  ∀b:T List, x:T.
. |  [x] = [u / (v @ b)]
. |  ⇒
. |  [u / v] = nil & b = [x] ∨ b = nil & [u / v] = [x]
. | by Auto THEN (SplitCons -1) THEN OrRight
. |
. |\
. | 1. T : Type
. | 2. T List
. | 3. u : T
. | 4. v : T List
. | 5. ∀b:T List, x:T.
. |    [x] = (v @ b) ⇒ v = nil & b = [x] ∨ b = nil & v = [x]
. | 6. b : T List
. | 7. x : T
. | 8. x = u
. | 9. nil = (v @ b)
. | ⊢  b = nil  by Obvious
. |
. .\
. . 1. T : Type
. . 2. T List
. . 3. u : T
. . 4. v : T List
. . 5. ∀b:T List, x:T.
. .    [x] = (v @ b) ⇒ v = nil & b = [x] ∨ b = nil & v = [x]
. . 6. b : T List
. . 7. x : T
. . 8. x = u
. . 9. nil = (v @ b)
. . ⊢  [u / v] = [x]  by Analyze THEN Obvious
END
```