```
l_member_decomp  THM

⊢  ∀T:Type, s:T List, z:T.
|  (z ∈ s) ⇒ (∃s1,s2:T List. s = (s1 @ [z / s2]))
| by InductionOnList
|
|\
| 1. T : Type
| 2. T List
| 3. z : T
| 4. (z ∈ nil)
| ⊢  ∃s1,s2:T List. nil = (s1 @ [z / s2])  by Obvious
|
.\
. 1. T : Type
. 2. T List
. 3. u : T
. 4. v : T List
. 5. ∀z:T. (z ∈ v) ⇒ (∃s1,s2:T List. v = (s1 @ [z / s2]))
. 6. z : T
. 7. (z ∈ [u / v])
. ⊢  ∃s1,s2:T List. [u / v] = (s1 @ [z / s2])
. | by (RWO
. |     Thm* ∀l:T List, a,x:T. (x ∈ [a / l]) ⟺ x = a ∨ (x ∈ l)
. |     {cons_member}
. |     -1)
. |    THEN
. |    SplitOrHyps
. |
. |\
. | 1. T : Type
. | 2. T List
. | 3. u : T
. | 4. v : T List
. | 5. ∀z:T. (z ∈ v) ⇒ (∃s1,s2:T List. v = (s1 @ [z / s2]))
. | 6. z : T
. | 7. z = u
. | ⊢  ∃s1,s2:T List. [u / v] = (s1 @ [z / s2])
. |   by (InstConcl [nil;v]) THEN (Reduce 0) THEN Analyze
. |
. .\
. . 1. T : Type
. . 2. T List
. . 3. u : T
. . 4. v : T List
. . 5. ∀z:T. (z ∈ v) ⇒ (∃s1,s2:T List. v = (s1 @ [z / s2]))
. . 6. z : T
. . 7. (z ∈ v)
. . ⊢  ∃s1,s2:T List. [u / v] = (s1 @ [z / s2])
. .   by (FwdThru -3 [-1]) THEN ExRepD THEN (InstConcl [[u / s1];s2])
. .      THEN
. .      (Reduce 0)
. .      THEN
. .      Analyze
END
```