```
member_iff_sublist  THM

⊢  ∀T:Type, x:T, L:T List. (x ∈ L) ⟺ [x] ⊆ L
| by Unfolds ``l_member sublist`` 0 THEN Auto'

|    THEN Try (Unfold `label` 0 THEN Reduce 0 THEN Auto')

|    THEN All Reduce THEN ExRepD
|
|\
| 1. T : Type
| 2. x : T
| 3. L : T List
| 4. i : ℕ
| 5. i<||L||
| 6. x = L[i]
| ⊢  ∃f:(ℕ1→ℕ||L||). increasing(f;1) & (∀j:ℕ1. [x][j] = L[(f(j))])
| | by InstConcl [λx.i] THEN Auto'
|
| |    THEN Try (Unfold `label` 0 THEN Reduce 0 THEN Auto')
| |
| |\
| | 1. T : Type
| | 2. x : T
| | 3. L : T List
| | 4. i : ℕ
| | 5. i<||L||
| | 6. x = L[i]
| | ⊢  increasing(λx.i;1)
| |   by Unfold `increasing` 0 THEN Reduce 0 THEN Auto'
| |
| .\
| . 1. T : Type
| . 2. x : T
| . 3. L : T List
| . 4. i : ℕ
| . 5. i<||L||
| . 6. x = L[i]
| . 7. j : ℕ1
| . ⊢  [x][j] = L[((λx.i)(j))]
| .   by AssertBY j = 0 Auto' THEN HypSubstSq (-1) 0 THEN
| .
| .      Reduce 0 THEN Auto'
|
.\
. 1. T : Type
. 2. x : T
. 3. L : T List
. 4. f : ℕ1→ℕ||L||
. 5. increasing(f;1)
. 6. ∀j:ℕ1. [x][j] = L[(f(j))]
. ⊢  ∃i:ℕ. i<||L|| & x = L[i]  by InstConcl [f(0)] THEN Auto'
. |
. 1. T : Type
. 2. x : T
```

```
. 3. L : T List
. 4. f : ℕ1→ℕ||L||
. 5. increasing(f;1)
. 6. ∀j:ℕ1. [x][j] = L[(f(j))]
. ⊢  x = L[(f(0))]  by InstHyp [0] (-1) THEN Reduce (-1) THEN Auto'
END
```