```
l_tricotomy  THM

⊢  ∀T:Type, x,y:T, L:T List.
|  (x ∈ L) ⇒ (y ∈ L) ⇒ x = y ∨ x before y ∈ L ∨ y before x ∈ L
| by Unfolds ‘‘l_member l_before‘‘ 0 THEN Auto THEN ExRepD
|
|
1. T : Type
2. x : T
3. y : T
4. L : T List
5. i1 : ℕ
6. i1<||L||
7. x = L[i1]
8. i : ℕ
9. i<||L||
10. y = L[i]
⊢  x = y ∨ [x; y] ⊆ L ∨ [y; x] ⊆ L
| by Tricotomy i1 i
|   THENL [Sel 1 (D 0) THENM Sel 2 (D 0);

|              Sel 1 (D 0) THENM Sel 1 (D 0);

|              Sel 2 (D 0)] THEN Auto
|
|\
| 1. T : Type
| 2. x : T
| 3. y : T
| 4. L : T List
| 5. i1 : ℕ
| 6. i1<||L||
| 7. x = L[i1]
| 8. i : ℕ
| 9. i<||L||
| 10. y = L[i]
| 11. i1<i
| ⊢  [x; y] ⊆ L
| | by Unfold ‘sublist‘ 0 THEN Reduce 0 THEN InstConcl
| [λj.if j=₂0→
| |                                                          i1
| |                                                      else i fi
| |    ] THEN Reduce 0 THEN Auto’
| |
| |    THEN Try (Unfold ‘increasing‘ 0 THEN Reduce 0 THEN Auto) THEN
| |
| |    Repeat (SplitOnConclITE THEN Auto’)
| |
| |
| |\
| | .....truecase .....
| | 1. T : Type
| | 2. x : T
| | 3. y : T
| | 4. L : T List
```

```
| | 5. i1 : ℕ
| | 6. i1<||L||
| | 7. x = L[i1]
| | 8. i : ℕ
| | 9. i<||L||
| | 10. y = L[i]
| | 11. i1<i
| | 12. j : ℕ2
| | 13. j = 0
| | ⊢  [x; y][j] = L[i1]
| |   by HypSubstSq (-1) 0 THEN Reduce 0 THEN Auto
| |
| .\
| . .....falsecase .....
| . 1. T : Type
| . 2. x : T
| . 3. y : T
| . 4. L : T List
| . 5. i1 : ℕ
| . 6. i1<||L||
| . 7. x = L[i1]
| . 8. i : ℕ
| . 9. i<||L||
| . 10. y = L[i]
| . 11. i1<i
| . 12. j : ℕ2
| . 13. ¬j = 0
| . ⊢  [x; y][j] = L[i]  by Subst' j = 1 0 THEN Reduce 0 THEN Auto
|
|\
| 1. T : Type
| 2. x : T
| 3. y : T
| 4. L : T List
| 5. i1 : ℕ
| 6. i1<||L||
| 7. x = L[i1]
| 8. i : ℕ
| 9. i<||L||
| 10. y = L[i]
| 11. i1 = i
| ⊢  x = y
|   by WeakSubstFor x 0 THENM WeakSubstFor y 0  THENM EqCD THEN Auto
|
.\
. 1. T : Type
. 2. x : T
. 3. y : T
. 4. L : T List
. 5. i1 : ℕ
. 6. i1<||L||
. 7. x = L[i1]
. 8. i : ℕ
. 9. i<||L||
. 10. y = L[i]
```

```
. 11. i<i1
. ⊢  [y; x] ⊆ L
. | by Unfold ‘sublist‘ 0 THEN Reduce 0 THEN InstConcl
. [λj.
. |    if j=₂0→ i else i1 fi] THEN Reduce 0 THEN Auto’
.
. |    THEN Try (Unfold ‘label‘ 0 THEN Reduce 0 THEN Auto)
. |
. |\
. | 1. T : Type
. | 2. x : T
. | 3. y : T
. | 4. L : T List
. | 5. i1 : ℕ
. | 6. i1<||L||
. | 7. x = L[i1]
. | 8. i : ℕ
. | 9. i<||L||
. | 10. y = L[i]
. | 11. i<i1
. | ⊢  increasing(λj.if j=₂0→ i else i1 fi;2)
. |   by Unfold ‘increasing‘ 0 THEN Reduce 0 THEN Auto THEN
. |
. |      SplitOnConclITE THEN Auto’
. | THEN SplitOnConclITE THEN Auto’
. |
. .\
. . 1. T : Type
. . 2. x : T
. . 3. y : T
. . 4. L : T List
. . 5. i1 : ℕ
. . 6. i1<||L||
. . 7. x = L[i1]
. . 8. i : ℕ
. . 9. i<||L||
. . 10. y = L[i]
. . 11. i<i1
. . 12. j : ℕ2
. . ⊢  [y; x][j] = L[if j=₂0→ i else i1 fi]
. . | by SplitOnConclITE THEN Auto’
. . |
. . |
. . |\
. . | .....truecase .....
. . | 1. T : Type
. . | 2. x : T
. . | 3. y : T
. . | 4. L : T List
. . | 5. i1 : ℕ
. . | 6. i1<||L||
. . | 7. x = L[i1]
. . | 8. i : ℕ
. . | 9. i<||L||
. . | 10. y = L[i]
```

```
. . | 11. i<i1
. . | 12. j : ℕ2
. . | 13. j = 0
. . | ⊢  [y; x][j] = L[i]
. . |   by HypSubstSq (-1) 0 THEN Reduce 0 THEN Auto
. . |
. . .\
. . . .....falsecase .....
. . . 1. T : Type
. . . 2. x : T
. . . 3. y : T
. . . 4. L : T List
. . . 5. i1 : ℕ
. . . 6. i1<||L||
. . . 7. x = L[i1]
. . . 8. i : ℕ
. . . 9. i<||L||
. . . 10. y = L[i]
. . . 11. i<i1
. . . 12. j : ℕ2
. . . 13. ¬j = 0
. . . ⊢  [y; x][j] = L[i1]
. . .   by RWO "select_cons_tl" 0 THEN Auto'
. . .
. . .      THEN Try (Unfold `label` 0 THEN Reduce 0 THEN Auto)
. . .
. . .      THEN RWO "select_cons_hd" 0 THEN Auto'
. . .
END
```