```
sublist_transitivity  THM

⊢  ∀T:Type, L1,L2,L3:T List. L1 ⊆ L2 ⇒ L2 ⊆ L3 ⇒ L1 ⊆ L3
| by Auto THEN All (Unfold `sublist`) THEN ExRepD
THEN InstConcl [f
|     o f1] THEN Auto
THEN Try (Complete Auto')
THEN Reduce 0
|
|\
| 1. T : Type
| 2. L1 : T List
| 3. L2 : T List
| 4. L3 : T List
| 5. f1 : ℕ||L1||→ℕ||L2||
| 6. increasing(f1;||L1||)
| 7. ∀j:ℕ||L1||. L1[j] = L2[(f1(j))]
| 8. f : ℕ||L2||→ℕ||L3||
| 9. increasing(f;||L2||)
| 10. ∀j:ℕ||L2||. L2[j] = L3[(f(j))]
| ⊢  increasing(f o f1;||L1||)
|   by BackThruLemma `compose_increasing` THEN Auto'
|
.\
. 1. T : Type
. 2. L1 : T List
. 3. L2 : T List
. 4. L3 : T List
. 5. f1 : ℕ||L1||→ℕ||L2||
. 6. increasing(f1;||L1||)
. 7. ∀j:ℕ||L1||. L1[j] = L2[(f1(j))]
. 8. f : ℕ||L2||→ℕ||L3||
. 9. increasing(f;||L2||)
. 10. ∀j:ℕ||L2||. L2[j] = L3[(f(j))]
. 11. j : ℕ||L1||
. ⊢  L1[j] = L3[(f(f1(j)))]  by InstHyp [f1(j)] (-2) THENA Auto
. |
. 1. T : Type
. 2. L1 : T List
. 3. L2 : T List
. 4. L3 : T List
. 5. f1 : ℕ||L1||→ℕ||L2||
. 6. increasing(f1;||L1||)
. 7. ∀j:ℕ||L1||. L1[j] = L2[(f1(j))]
. 8. f : ℕ||L2||→ℕ||L3||
. 9. increasing(f;||L2||)
. 10. ∀j:ℕ||L2||. L2[j] = L3[(f(j))]
. 11. j : ℕ||L1||
. 12. L2[(f1(j))] = L3[(f(f1(j)))]
. ⊢  L1[j] = L3[(f(f1(j)))]  by InstHyp [j] (-6) THEN Auto
END
```