```
P_no_dup_strong_safety  THM

⊢  ∀E:EventStruct. R_strong_safety(E) preserves No-dup-deliver(E)
| by Unfolds ``preserved_by P_no_dup R_strong_safety`` 0

|    THEN Unfold `sublist` 0
THEN Reduce 0
THEN Auto'
THEN ExRepD
|
THEN InstHyp [f(i);f(j)] 4 THEN Auto

|    THEN AllHyps (\h. InstHyp [j
|    ] h THENA Auto THEN RevHypSubst (-1) 0 THEN Auto')

|    THEN AllHyps (\h. InstHyp [i
|    ] h THENA Auto THEN RevHypSubst (-1) 0 THEN Auto')
|
1. E : EventStruct
2. x : |E| List
3. y : |E| List
4. ∀i,j:ℕ||x||.
   ¬is-send(E)(x[i]) ⇒
   ¬is-send(E)(x[j]) ⇒
   (x[j] =msg=(E) x[i]) ⇒ loc(E)(x[i]) = loc(E)(x[j]) ⇒ i = j
5. f : ℕ||y||→ℕ||x||
6. increasing(f;||y||)
7. ∀j:ℕ||y||. y[j] = x[(f(j))]
8. i : ℕ||y||
9. j : ℕ||y||
10. ¬is-send(E)(y[i])
11. ¬is-send(E)(y[j])
12. y[j] =msg=(E) y[i]
13. loc(E)(y[i]) = loc(E)(y[j])
14. f(i) = f(j)
15. y[j] = x[(f(j))]
16. y[i] = x[(f(i))]
⊢  i = j
  by AllHyps (\h. FwdThruLemma `increasing_inj` [h] THENA Auto')

     THEN Unfold `inject` (-1) THEN BackThruHyp (-1) THEN Auto'
END
```