```
no_dup_fusion  THM

⊢  ∀E:TaggedEventStruct. Tag-by-msg(E) fuses No-dup-deliver(E)
| by Unfold ‘fusion_condition‘ 0 THEN Unfold ‘str_trace‘ 0 THEN Auto

|    THEN  Try (Fold ‘trace_property‘ 0 THEN Auto)
|
1. E : TaggedEventStruct
2. tr : |E| List
3. ∀m:Label. No-dup-deliver(E)(<tr>_m)
4. Tag-by-msg(E)(tr)
⊢  No-dup-deliver(E)(tr)
| by RWO "P_no_dup_iff" 0 THEN Auto
THEN AllHyps (InstHyp [tag(E)(x)
|    ]) THENA Auto
THEN RWO "P_no_dup_iff" (-1) THENA Auto

|    THEN InstHyp [x;y] (-1) THENA Auto
|
1. E : TaggedEventStruct
2. tr : |E| List
3. ∀m:Label. No-dup-deliver(E)(<tr>_m)
4. Tag-by-msg(E)(tr)
5. x : |E|
6. y : |E|
7. ¬is-send(E)(x)
8. ¬is-send(E)(y)
9. y =msg=(E) x
10. loc(E)(x) = loc(E)(y)
11. ∀x1,y:|E|.
|   ¬is-send(E)(x1) ⇒
|   ¬is-send(E)(y) ⇒
|   (y =msg=(E) x1) ⇒
|   loc(E)(x1) = loc(E)(y) ⇒ ¬sublist(|E|;[x1; y];<tr>_tag(E)(x))
12. ¬sublist(|E|;[x; y];<tr>_tag(E)(x))
⊢  ¬sublist(|E|;[x; y];tr)
| by ParallelOp (-1) THEN Unfold ‘tag_sublist‘ 0

|    THEN BackThruLemma ‘sublist_filter‘ THEN Auto

|    THEN Unfold ‘l_all‘ 0 THEN Reduce 0 THEN Auto
|
1. E : TaggedEventStruct
2. tr : |E| List
3. ∀m:Label. No-dup-deliver(E)(<tr>_m)
4. Tag-by-msg(E)(tr)
5. x : |E|
6. y : |E|
7. ¬is-send(E)(x)
8. ¬is-send(E)(y)
9. y =msg=(E) x
10. loc(E)(x) = loc(E)(y)
11. ∀x1,y:|E|.
|   ¬is-send(E)(x1) ⇒
|   ¬is-send(E)(y) ⇒
```

```
|    (y =msg=(E) x1) ⇒
|    loc(E)(x1) = loc(E)(y) ⇒ ¬sublist(|E|;[x1; y];<tr>_tag(E)(x))
12. sublist(|E|;[x; y];tr)
13. x1 : |E|
14. (x1 ∈ [x; y])
⊢  tag(E)(x1) =₁ tag(E)(x)
| by RWW "cons_member" (-1) THENM RWW "nil_member" (-1) THENA Auto

|     THEN SplitOrHyps THEN Try Trivial THEN SubstFor x1 0 THEN Auto
|
|\
| 1. E : TaggedEventStruct
| 2. tr : |E| List
| 3. ∀m:Label. No-dup-deliver(E)(<tr>_m)
| 4. Tag-by-msg(E)(tr)
| 5. x : |E|
| 6. y : |E|
| 7. ¬is-send(E)(x)
| 8. ¬is-send(E)(y)
| 9. y =msg=(E) x
| 10. loc(E)(x) = loc(E)(y)
| 11. ∀x1,y:|E|.
|      ¬is-send(E)(x1) ⇒
|      ¬is-send(E)(y) ⇒
|      (y =msg=(E) x1) ⇒
|      loc(E)(x1) = loc(E)(y) ⇒ ¬sublist(|E|;[x1; y];<tr>_tag(E)(x))
| 12. sublist(|E|;[x; y];tr)
| 13. x1 : |E|
| 14. x1 = x
| ⊢  tag(E)(x) =₁ tag(E)(x)  by EqLblReflexive 0
|
.\
. 1. E : TaggedEventStruct
. 2. tr : |E| List
. 3. ∀m:Label. No-dup-deliver(E)(<tr>_m)
. 4. Tag-by-msg(E)(tr)
. 5. x : |E|
. 6. y : |E|
. 7. ¬is-send(E)(x)
. 8. ¬is-send(E)(y)
. 9. y =msg=(E) x
. 10. loc(E)(x) = loc(E)(y)
. 11. ∀x1,y:|E|.
. |    ¬is-send(E)(x1) ⇒
. |    ¬is-send(E)(y) ⇒
. |    (y =msg=(E) x1) ⇒
. |    loc(E)(x1) = loc(E)(y) ⇒ ¬sublist(|E|;[x1; y];<tr>_tag(E)(x))
. 12. sublist(|E|;[x; y];tr)
. 13. x1 : |E|
. 14. x1 = y
. ⊢  tag(E)(y) =₁ tag(E)(x)
. | by Unfold ‘sublist‘ (-3) THEN Reduce (-3) THEN ExRepD
.
. |     THEN InstHyp [0] (-3) THENA Auto THEN Reduce (-1)
.
```

```
. |    THEN InstHyp [1] (-4) THENA Auto THEN Reduce (-1)
.
. |    THEN AllHyps (\h. Unfold `P_tag_by_msg` h THEN Reduce h
.
. |              THEN InstHyp [f(0);f(1)] h THEN Auto)
. |
. |\
. | .....antecedent .....
. | 1. E : TaggedEventStruct
. | 2. tr : |E| List
. | 3. ∀m:Label. No-dup-deliver(E)(<tr>_m)
. | 4. ∀i,j:ℕ||tr||.
. |    (tr[i] =msg=(E) tr[j]) ⇒ tag(E)(tr[i]) = tag(E)(tr[j])
. | 5. x : |E|
. | 6. y : |E|
. | 7. ¬is-send(E)(x)
. | 8. ¬is-send(E)(y)
. | 9. y =msg=(E) x
. | 10. loc(E)(x) = loc(E)(y)
. | 11. ∀x1,y:|E|.
. |     ¬is-send(E)(x1) ⇒
. |     ¬is-send(E)(y) ⇒
. |     (y =msg=(E) x1) ⇒
. |     loc(E)(x1) = loc(E)(y) ⇒ ¬sublist(|E|;[x1; y];<tr>_tag(E)(x))
. | 12. f : ℕ2→ℕ||tr||
. | 13. increasing(f;2)
. | 14. ∀j:ℕ2. [x; y][j] = tr[(f(j))]
. | 15. x1 : |E|
. | 16. x1 = y
. | 17. x = tr[(f(0))]
. | 18. y = tr[(f(1))]
. | ⊢  tr[(f(0))] =msg=(E) tr[(f(1))]
. |   by SubstFor tr[(f(0))] 0 THEN SubstFor tr[(f(1))] 0
. |
. |      THEN InstLemma `event_msg_eq_equiv` [E] THENA Auto
. |
. |      THEN UseEquiv
. |
. .\
. . 1. E : TaggedEventStruct
. . 2. tr : |E| List
. . 3. ∀m:Label. No-dup-deliver(E)(<tr>_m)
. . 4. ∀i,j:ℕ||tr||.
. .    (tr[i] =msg=(E) tr[j]) ⇒ tag(E)(tr[i]) = tag(E)(tr[j])
. . 5. x : |E|
. . 6. y : |E|
. . 7. ¬is-send(E)(x)
. . 8. ¬is-send(E)(y)
. . 9. y =msg=(E) x
. . 10. loc(E)(x) = loc(E)(y)
. . 11. ∀x1,y:|E|.
. .     ¬is-send(E)(x1) ⇒
. .     ¬is-send(E)(y) ⇒

. .     (y =msg=(E) x1) ⇒
. .     loc(E)(x1) = loc(E)(y) ⇒ ¬sublist(|E|;[x1; y];<tr>_tag(E)(x))
```

```
. . 12. f : ℕ2→ℕ||tr||
. . 13. increasing(f;2)
. . 14. ∀j:ℕ2. [x; y][j] = tr[(f(j))]
. . 15. x1 : |E|
. . 16. x1 = y
. . 17. x = tr[(f(0))]
. . 18. y = tr[(f(1))]
. . 19. tag(E)(tr[(f(0))]) = tag(E)(tr[(f(1))])
. . ⊢  tag(E)(y) =₁ tag(E)(x)
. .   by SubstFor y 0 THEN SubstFor x 0
. .
. .      THEN HypSubstSq (-1) 0 THEN EqLblReflexive 0
END
```