```
switch_inv_rel_closure  THM

⊢  ∀E:TaggedEventStruct, tr:|E| List, ls:ℕ||tr||.
|  switch_inv(E)(tr) ⇒
|  (∀i,j:ℕ||tr||.
|   (i (switchR(tr)^*) ls) ⇒
|   (j (switchR(tr)^*) ls) ⇒ tag(E)(tr[i]) = tag(E)(tr[j]))
| by RepeatFor 8 (D 0 THENA Auto)
|
1. E : TaggedEventStruct
2. tr : |E| List
3. ls : ℕ||tr||
4. switch_inv(E)(tr)
5. i : ℕ||tr||
6. j : ℕ||tr||
7. i (switchR(tr)^*) ls
8. j (switchR(tr)^*) ls
⊢  tag(E)(tr[i]) = tag(E)(tr[j])
| by InstLemma `rel_star_closure` [ℕ||tr||;switchR(tr);λi,j. tag(E)
|                                                           (tr[i])
|                                                           =
|                                                           tag(E)
|                                                           (tr[j])
|    ] THENA Auto

|    THEN Try (Unfolds ``trans refl``  0 THEN ReduceSOAps 0 THEN Auto)
|
THEN All  ReduceSOAps
THEN All Reduce
|
|\
| 1. E : TaggedEventStruct
| 2. tr : |E| List
| 3. ls : ℕ||tr||
| 4. switch_inv(E)(tr)
| 5. i : ℕ||tr||
| 6. j : ℕ||tr||
| 7. i (switchR(tr)^*) ls
| 8. j (switchR(tr)^*) ls
| 9. x : ℕ||tr||
| 10. y : ℕ||tr||
| 11. x switchR(tr) y
| ⊢  tag(E)(tr[x]) = tag(E)(tr[y])
| | by ...
| |
| |
| 1. E : TaggedEventStruct
| 2. tr : |E| List
| 3. ls : ℕ||tr||
| 4. switch_inv(E)(tr)
| 5. i : ℕ||tr||
| 6. j : ℕ||tr||
| 7. i (switchR(tr)^*) ls
| 8. j (switchR(tr)^*) ls
| 9. x : ℕ||tr||
```

```
| 10. y : ℕ||tr||
| 11. x switchR(tr) y
| ⊢  tag(E)(tr[x]) = tag(E)(tr[y])
|   by BackThruLemma ‘switch_inv_rel_same_tag‘ THEN Auto
|
.\
. 1. E : TaggedEventStruct
. 2. tr : |E| List
. 3. ls : ℕ||tr||
. 4. switch_inv(E)(tr)
. 5. i : ℕ||tr||
. 6. j : ℕ||tr||
. 7. i (switchR(tr)^*) ls
. 8. j (switchR(tr)^*) ls
. 9. ∀x,y:ℕ||tr||.
.    (x (switchR(tr)^*) y) ⇒ tag(E)(tr[x]) = tag(E)(tr[y]) ∨ x = y
. ⊢  tag(E)(tr[i]) = tag(E)(tr[j])
.   by InstHyp [i;ls] (-1) THENA Auto THEN SimpHyp (-1)
.
.      THEN InstHyp [j;ls] (-2) THENA Auto THEN SimpHyp (-1)
.
.      THEN SplitOrHyps THEN Auto
. THEN Try (SubstFor i 0 )
.
.      THEN Try (SubstFor j 0) THEN Auto
END
```