```
P_causal_iff  THM

⊢  ∀E:EventStruct, tr:|E| List.
|  Causal(E)(tr)
|  ⟺
|  (∀tr':|E| List.
|   tr' ≤ tr ⇒ (∀x∈tr'.(∃y∈tr'.is-send(E)(y) & (y =msg=(E) x))))
| by Unfolds ``P_causal l_all l_exists`` 0 THEN Unfold `l_member` 0

|    THEN Reduce 0 THEN Auto THEN ExRepD
|
|\
| 1. E : EventStruct
| 2. tr : |E| List
| 3. ∀i:ℕ||tr||.
| |  ∃j:ℕ||tr||. j≤i & is-send(E)(tr[j]) & (tr[j] =msg=(E) tr[i])
| 4. tr' : |E| List
| 5. tr' ≤ tr
| 6. x : |E|
| 7. i : ℕ
| 8. i<||tr'||
| 9. x = tr'[i]
| ⊢  ∃y:|E|.
| |  (∃i:ℕ. i<||tr'|| & y = tr'[i]) & is-send(E)(y) & (y =msg=(E) x)
| | by AssertBY ||tr'||≤||tr|| (BackThruLemma `iseg_length` THEN Auto)
| |
| |
| 1. E : EventStruct
| 2. tr : |E| List
| 3. ∀i:ℕ||tr||.
| |  ∃j:ℕ||tr||. j≤i & is-send(E)(tr[j]) & (tr[j] =msg=(E) tr[i])
| 4. tr' : |E| List
| 5. tr' ≤ tr
| 6. x : |E|
| 7. i : ℕ
| 8. i<||tr'||
| 9. x = tr'[i]
| 10. ||tr'||≤||tr||
| ⊢  ∃y:|E|.
| |  (∃i:ℕ. i<||tr'|| & y = tr'[i]) & is-send(E)(y) & (y =msg=(E) x)
| | by AllHyps (RWO "iseg_select") THENA Auto THEN ExRepD
|
| |    THEN InstHyp [i] 3 THENA Auto' THEN ExRepD THEN InstConcl [
| |    tr[j]] THEN Auto
| |
| |\
| | 1. E : EventStruct
| | 2. tr : |E| List
| | 3. ∀i:ℕ||tr||.
| |    ∃j:ℕ||tr||. j≤i & is-send(E)(tr[j]) & (tr[j] =msg=(E) tr[i])
| | 4. tr' : |E| List
| | 5. ||tr'||≤||tr||
| | 6. ∀i:ℕ. i<||tr'|| ⇒ tr'[i] = tr[i]
| | 7. x : |E|
| | 8. i : ℕ
```

```
| | 9. i<||tr'||
| | 10. x = tr'[i]
| | 11. ||tr'||≤||tr||
| | 12. j : ℕ||tr||
| | 13. j≤i
| | 14. is-send(E)(tr[j])
| | 15. tr[j] =msg=(E) tr[i]
| | ⊢  ∃i:ℕ. i<||tr'|| & tr[j] = tr'[i]
| |   by InstConcl [j] THEN Auto THEN Symmetry THEN EasyHyp
| |
| .\
| . 1. E : EventStruct
| . 2. tr : |E| List
| . 3. ∀i:ℕ||tr||.
| .    ∃j:ℕ||tr||. j≤i & is-send(E)(tr[j]) & (tr[j] =msg=(E) tr[i])
| . 4. tr' : |E| List
| . 5. ||tr'||≤||tr||
| . 6. ∀i:ℕ. i<||tr'|| ⇒ tr'[i] = tr[i]
| . 7. x : |E|
| . 8. i : ℕ
| . 9. i<||tr'||
| . 10. x = tr'[i]
| . 11. ||tr'||≤||tr||
| . 12. j : ℕ||tr||
| . 13. j≤i
| . 14. is-send(E)(tr[j])
| . 15. tr[j] =msg=(E) tr[i]
| . ⊢  tr[j] =msg=(E) x
| .   by Subst x = tr[i] 0 THEN Auto THEN SubstFor x 0 THEN EasyHyp
|
.\
. 1. E : EventStruct
. 2. tr : |E| List
. 3. ∀tr':|E| List.
. |  tr' ≤ tr ⇒
. |  (∀x:|E|.
. |   (∃i:ℕ. i<||tr'|| & x = tr'[i]) ⇒
. |   (∃y:|E|.
. |    (∃i:ℕ. i<||tr'|| & y = tr'[i])
. |    & is-send(E)(y)
. |    & (y =msg=(E) x)))
. 4. i : ℕ||tr||
. ⊢  ∃j:ℕ||tr||. j≤i & is-send(E)(tr[j]) & (tr[j] =msg=(E) tr[i])
. | by InstHyp [firstn(i+1;tr);tr[i]] 3 THEN Auto
. |
. |\
. | .....antecedent .....
. | 1. E : EventStruct
. | 2. tr : |E| List
. | 3. ∀tr':|E| List.
. |    tr' ≤ tr ⇒
. |    (∀x:|E|.
. |     (∃i:ℕ. i<||tr'|| & x = tr'[i]) ⇒
. |     (∃y:|E|.
. |      (∃i:ℕ. i<||tr'|| & y = tr'[i])
```

```
. |      & is-send(E)(y)
. |      & (y =msg=(E) x)))
. | 4. i : ℕ||tr||
. | ⊢  firstn(i+1;tr) ≤ tr
. |   by RWO "firstn_is_iseg" 0 THENA Auto THEN InstConcl [i+1
. |      ] THEN Auto'
. |
. |\
. | .....antecedent .....
. | 1. E : EventStruct
. | 2. tr : |E| List
. | 3. ∀tr':|E| List.
. |    tr' ≤ tr ⇒
. |    (∀x:|E|.
. |     (∃i:ℕ. i<||tr'|| & x = tr'[i]) ⇒
. |     (∃y:|E|.
. |      (∃i:ℕ. i<||tr'|| & y = tr'[i])
. |      & is-send(E)(y)
. |      & (y =msg=(E) x)))
. | 4. i : ℕ||tr||
. | ⊢  ∃i@0:ℕ. i@0<||firstn(i+1;tr)|| & tr[i] = firstn(i+1;tr)[i@0]
. |   by InstConcl [i] THEN Auto
. |
. |      THEN Try (RWO "length_firstn" 0 THEN Auto)
. |
. |      THEN Try (RWO "select_firstn" 0 THEN Auto)
. |
. .\
. . 1. E : EventStruct
. . 2. tr : |E| List
. . 3. ∀tr':|E| List.
. . |  tr' ≤ tr ⇒
. . |  (∀x:|E|.
. . |   (∃i:ℕ. i<||tr'|| & x = tr'[i]) ⇒
. . |   (∃y:|E|.
. . |    (∃i:ℕ. i<||tr'|| & y = tr'[i])
. . |    & is-send(E)(y)
. . |    & (y =msg=(E) x)))
. . 4. i : ℕ||tr||
. . 5. ∃y:|E|.
. . |  (∃i@0:ℕ. i@0<||firstn(i+1;tr)|| & y = firstn(i+1;tr)[i@0])
. . |  & is-send(E)(y)
. . |  & (y =msg=(E) tr[i])
. . ⊢  ∃j:ℕ||tr||. j≤i & is-send(E)(tr[j]) & (tr[j] =msg=(E) tr[i])
. . | by ExRepD
. . THEN AllHyps (\h. RWO "length_firstn" h THENA Auto)
. . |
. . THEN AllHyps (\h. RWO "select_firstn" h THENA Auto)
. . |
. . 1. E : EventStruct
. . 2. tr : |E| List
. . 3. ∀tr':|E| List.
. .    tr' ≤ tr ⇒
. .    (∀x:|E|.
. .     (∃i:ℕ. i<||tr'|| & x = tr'[i]) ⇒
```

```
. .     (∃y:|E|.
. .      (∃i:ℕ. i<||tr'|| & y = tr'[i])
. .      & is-send(E)(y)
. .      & (y =msg=(E) x)))
. . 4. i : ℕ||tr||
. . 5. y : |E|
. . 6. i@0 : ℕ
. . 7. i@0<i+1
. . 8. y = tr[i@0]
. . 9. is-send(E)(y)
. . 10. y =msg=(E) tr[i]
. . ⊢  ∃j:ℕ||tr||. j≤i & is-send(E)(tr[j]) & (tr[j] =msg=(E) tr[i])
. .   by InstConcl [i@0] THEN Auto'
. . THEN SubstFor tr[i@0] 0 THEN Auto
. .
END
```