```
covers_pred_addprime  THM

⊢  ∀Q:Fmla, A:ioa{i:l}(). covers_pred(A;(Q)') ⟺ covers_pred(A;Q)
| by Unfolds ``covers_pred pred_addprime`` 0
THEN Auto

|    THEN Try (Unfold `pred` 0 THEN Auto THEN Fold `pred` 0 THEN Auto)
|
THEN All (RW ColMemberC)
THEN Try BackThruSomeHyp

|    THEN All (Unfold `pred_mentions`)
THEN All (RW ColMemberC)

|    THEN ExRepD
THEN Try (InstConcl [r@0] THEN Auto

|              THEN Try (Fold `pred` 0 THEN Auto)

|              THEN HypSubst(-2) (-1) THENA Auto)

|    THEN Try (InstConcl [(r)'] THEN Auto

|             THEN Try (AutoInstConcl [] THEN Auto)

|             THEN Try (Fold `pred` 0) THEN Auto)

|    THEN All (Unfolds ``rel_addprime rel_mentions``)
THEN ExRepD

|    THEN InstConcl [i]
THEN All Reduce
THEN Auto

|    THEN All (\i.RWW "map_length_nat" i THEN Auto)

|    THEN All (\i. RWW "map_select" i THEN Auto THEN Reduce i)
.
|
|\
| 1. Q : Fmla
| 2. A : ioa{i:l}()
| 3. ∀x:Label.
|    (∃r:rel().
|     (∃r@0:rel().
|      r@0 ∈ Q & r = mk_rel(r@0.name, map(λt.(t)';r@0.args)))
|     & (∃i:ℕ. i<||r.args|| & (x ∈ term_vars(r.args[i]))))
|    ⇒
|    covers_var(A;x)
| 4. x : Label
| 5. r : rel()
| 6. r ∈ Q
| 7. i : ℕ
| 8. i<||r.args||
| 9. (x ∈ term_vars(r.args[i]))
```

```
| ⊢  (x ∈ term_vars((r.args[i])'))
|   by InstLemma ‘term_vars_addprime‘ [r.args[i]]
|
|         THENM RevHypSubst (-1) (-2) THEN Auto
|
|         THEN BackThruLemma ‘term_primed_vars_term_vars‘ THEN Auto⋅
|
.\
. 1. Q : Fmla
. 2. A : ioa{i:l}()
. 3. ∀x:Label.
.    (∃r:rel().
.     r ∈ Q & (∃i:ℕ. i<||r.args|| & (x ∈ term_vars(r.args[i]))))
.    ⇒
.    covers_var(A;x)
. 4. x : Label
. 5. r : rel()
. 6. r@0 : rel()
. 7. r@0 ∈ Q
. 8. r = mk_rel(r@0.name, map(λt.(t)';r@0.args))
. 9. i : ℕ
. 10. i<||r@0.args||
. 11. (x ∈ term_vars((r@0.args[i])'))
. ⊢  (x ∈ term_vars(r@0.args[i]))
.   by MoveToConcl (-1) THEN GenConcl r@0.args[i] = t
.       THENA Auto THEN Thin (-1)
.
.      ...
.    THEN Try (Fold ‘addprime‘ 0)
.
.         THEN RWO "member_append" 0 THEN Auto
.
.         THEN ParallelOp (-1) THEN EasyHyp⋅
END
```