```
firstn_append  THM

⊢  ∀L1,L2:Top List, n:ℕ(||L1||+1). firstn(n;L1 @ L2) ~ firstn(n;L1)
| by InductionOnList THEN RecUnfold ‘firstn‘ 0 THEN Reduce 0
|
|\
| 1. Top List
| ⊢  ∀L2:Top List, n:ℕ1.
|    (Case of L2
|       nil → nil
|       a.as’ → if 0<₂n→ [a / firstn(n-1;as’)] else nil fi) ~ nil
|   by Auto THEN Analyze -2 THEN Reduce 0 THEN Try Trivial
|      THEN
|      SplitOnConclITE
|
.\
. 1. Top List
. 2. u : Top
. 3. v : Top List
. 4. ∀L2:Top List, n:ℕ(||v||+1). firstn(n;v @ L2) ~ firstn(n;v)
. ⊢  ∀L2:Top List, n:ℕ(||v||+1+1).
. |  if 0<₂n→ [u / firstn(n-1;v @ L2)] else nil fi ~ if 0<₂n→
. |                                                      [u /
. |                                                       firstn(n
. |                                                       -1;v)]
. |                                                   else nil fi
. | by UnivCD THEN SplitOnConclITE THEN Analyze THEN Try Trivial
. |
. 1. Top List
. 2. Top
. 3. v : Top List
. 4. ∀L2:Top List, n:ℕ(||v||+1). firstn(n;v @ L2) ~ firstn(n;v)
. 5. L2 : Top List
. 6. n : ℕ(||v||+1+1)
. 7. 0<n
. ⊢  firstn(n-1;v @ L2) ~ firstn(n-1;v)  by BackThru 4
END
```