```
firstn_lem1  THM

⊢  ∀x:ℕ, as:T*. x<||as|| ⇒ firstn(x+1;as) = (firstn(x;as) @ [as[x]])
| by (Analyze 0) THEN (Analyze 0)
|
1. T : Type
2. x : ℕ
⊢  ∀as:T*. x<||as|| ⇒ firstn(x+1;as) = (firstn(x;as) @ [as[x]])
| by NatInd 2 THEN RepD
|
|\
| 2. as : T*
| 3. 0<||as||
| ⊢  firstn(0+1;as) = (firstn(0;as) @ [as[0]])
| | by RecUnfold `firstn` 0 THEN Reduce 0 THEN ListInd 2
| |
| |\
| | 2. T*
| | ⊢  0<||nil|| ⇒
| |    (Case of nil; nil → nil ; a.as' → a.firstn(0;as'))
| |    =
| |    ((Case of nil; nil → nil ; a.as' → nil) @ [nil[0]])
| |   by Reduce 0
| |
| .\
| . 3. u : T
| . 4. v : T*
| . 5. 0<||v|| ⇒
| . |  (Case of v; nil → nil ; a.as' → a.firstn(0;as'))
| . |  =
| . |  ((Case of v; nil → nil ; a.as' → nil) @ [v[0]])
| . ⊢  0<||u.v|| ⇒
| . |  (Case of u.v; nil → nil ; a.as' → a.firstn(0;as'))
| . |  =
| . |  ((Case of u.v; nil → nil ; a.as' → nil) @ [(u.v)[0]])
| . | by Thin -1 THEN Reduce 0 THEN Analyze 0
| . |
| . 5. 0<||v||+1
| . ⊢  u.firstn(0;v) = [u]  by Analyze
| . |
| . ⊢  firstn(0;v) = nil
| .   by Thin -1 THEN ListInd 4 THEN RecUnfold `firstn` 0
| .      THEN
| .      Reduce 0
|
.\
. 2. x : ℤ
. 3. 0<x
. 4. ∀as:T*.
. |  x-1<||as|| ⇒ firstn(x-1+1;as) = (firstn(x-1;as) @ [as[(x-1)]])
. 5. as : T*
. 6. x<||as||
. ⊢  firstn(x+1;as) = (firstn(x;as) @ [as[x]])  by ListInd 5
. |
. |\
```

```
. | 5. T*
. | ⊢  x<||nil|| ⇒ firstn(x+1;nil) = (firstn(x;nil) @ [nil[x]])
. |   by Reduce 0
. |
. .\
. . 6. u : T
. . 7. v : T*
. . 8. x<||v|| ⇒ firstn(x+1;v) = (firstn(x;v) @ [v[x]])
. . ⊢  x<||u.v|| ⇒ firstn(x+1;u.v) = (firstn(x;u.v) @ [(u.v)[x]])
. . | by Thin -1 THEN Reduce 0 THEN Analyze 0
. . |
. . 8. x<||v||+1
. . ⊢  firstn(x+1;u.v) = (firstn(x;u.v) @ [(u.v)[x]])
. . | by RecUnfold `firstn` 0 THEN Reduce 0
. . |
. . ⊢  if 0<₂x+1→ u.firstn(x+1-1;v) else nil fi
. . |  =
. . |  (if 0<₂x→ u.firstn(x-1;v) else nil fi @ [(u.v)[x]])
. . | by SplitOnConclITE
. . |
. . .....truecase .....
. . 9. 0<x+1
. . ⊢  u.firstn(x+1-1;v)
. . |  =
. . |  (if 0<₂x→ u.firstn(x-1;v) else nil fi @ [(u.v)[x]])
. . | by SplitOnConclITE
. . |
. . .....truecase .....
. . 10. 0<x
. . ⊢  u.firstn(x+1-1;v) = ((u.firstn(x-1;v)) @ [(u.v)[x]])
. . | by (ArithSimp 0) THEN (ArithSimp 4)
. . |
. . |\
. . | ⊢  x<||u.v||  by Reduce 0
. . |
. . .\
. . . 4. ∀as:T*.
. . . |  -1+x<||as|| ⇒
. . . |  firstn(x;as) = (firstn(-1+x;as) @ [as[(-1+x)]])
. . . 5. T*
. . . 6. u : T
. . . 7. v : T*
. . . 8. x<||v||+1
. . . 9. 0<x+1
. . . 10. 0<x
. . . ⊢  u.firstn(x;v) = ((u.firstn(-1+x;v)) @ [(u.v)[x]])
. . . | by Reduce 0 THEN Analyze
. . . |
. . . |\
. . . | ⊢  u = u  by Auto
. . . |
. . . .\
. . . . ⊢  firstn(x;v) = (firstn(-1+x;v) @ [(u.v)[x]])
. . . . | by (InstHyp [v] 4) THEN (RWH (HypC -1) 0)
. . . . |
```

```
. . . . |\
. . . . | 11. firstn(x;v) = (firstn(-1+x;v) @ [v[(-1+x)]])
. . . . | ⊢  x<||u.v||  by Reduce 0
. . . . |
. . . . .\
. . . . . 11. firstn(x;v) = (firstn(-1+x;v) @ [v[(-1+x)]])
. . . . . ⊢  (firstn(-1+x;v) @ [v[(-1+x)]])
. . . . . |  =
. . . . . |  (firstn(-1+x;v) @ [(u.v)[x]])  by Analyze
. . . . . |
. . . . . ⊢  [v[(-1+x)]] = [(u.v)[x]]  by Unfold ‘select‘ 0
. . . . . |
. . . . . ⊢  [hd(nth_tl(-1+x;v))] = [hd(nth_tl(x;u.v))]
. . . . . | by RWH (AddrC [3] (RecUnfoldC ‘nth_tl‘)) 0
. . . . . |
. . . . . ⊢  [hd(nth_tl(-1+x;v))]
. . . . . |  =
. . . . . |  [hd(if x≤₂0→ u.v else nth_tl(x-1;tl((u.v))) fi)]
. . . . . | by SplitOnConclITE
. . . . . |
. . . . . .....falsecase .....
. . . . . 12. 0<x
. . . . . ⊢  [hd(nth_tl(-1+x;v))] = [hd(nth_tl(x-1;tl((u.v))))]
. . . . .   by ArithSimp 0 THEN Reduce 0
END
```