```
lpower_alt  THM

⊢  ∀L:Alph*, n:ℕ. (L↑n) = if n=₂0⟶ nil else L @ (L↑n-1) fi  by UnivCD
|
1. Alph : Type
2. L : Alph*
3. n : ℕ
⊢  (L↑n) = if n=₂0⟶ nil else L @ (L↑n-1) fi  by NatInd 3
|
|\
| .....basecase .....
| 2. L : Alph*
| ⊢  (L↑0) = if 0=₂0⟶ nil else L @ (L↑0-1) fi  by Reduce 0
| |
| ⊢  (L↑0) = nil  by RecCaseSplit `lpower`
|
.\
. .....upcase .....
. 3. n : ℤ
. 4. 0<n
. 5. (L↑n-1) = if n-1=₂0⟶ nil else L @ (L↑n-1-1) fi
. ⊢   by SplitOnConclITE
. |
. |\
. | .....truecase .....
. | 6. n = 0
. | ⊢  (L↑n) = nil  by Auto
. |
. .\
. . .....falsecase .....
. . 6. ¬n = 0
. . ⊢  (L↑n) = (L @ (L↑n-1))
. . | by RWH (AddrC [3;2] (RecUnfoldC `lpower`)) 0
. . |
. . ⊢  (L↑n) = (L @ if n-1=₂0⟶ nil else (L↑n-1-1) @ L fi)
. . | by SplitOnConclITE
. . |
. . |\
. . | .....truecase .....
. . | 7. n-1 = 0
. . | ⊢  (L↑n) = (L @ nil)
. . | | by RWH
. . | |    (LemmaC Thm* ∀as:T*. (as @ nil) = as    {append_back_nil})
. . | |    0
. . | |
. . | ⊢  (L↑n) = L  by RecCaseSplit `lpower`
. . | |
. . | |\
. . | | .....truecase .....
. . | | 8. n = 0
. . | | ⊢  nil = L  by Auto
. . | |
. . | .\
. . | . .....falsecase .....
. . | . 8. ¬n = 0
```

```
. . | . ⊢  ((L↑n-1) @ L) = L  by RecCaseSplit ‘lpower‘
. . | . |
. . | . |\
. . | . | 9. n-1 = 0
. . | . | ⊢  L = L  by Auto
. . | . |
. . | . .\
. . | . . .....falsecase .....
. . | . . 9. ¬n-1 = 0
. . | . . ⊢  (((L↑n-1-1) @ L) @ L) = L  by Auto
. . |
. . .\
. . . .....falsecase .....
. . . 7. ¬n-1 = 0
. . . ⊢  (L↑n) = (L @ (L↑n-1-1) @ L)
. . . | by Assert ((L↑n-1) = (L @ (L↑n-1-1)))
. . . |
. . . |\
. . . | .....assertion .....
. . . | ⊢  (L↑n-1) = (L @ (L↑n-1-1))
. . . |   by RWH (HypC 5) 0 THEN SplitOnConclITE
. . . |
. . . .\
. . . . 8. (L↑n-1) = (L @ (L↑n-1-1))
. . . . ⊢
. . . . | by RWH
. . . . |    (RevLemmaC
. . . . |     Thm* ∀as,bs,cs:T*. ((as @ bs) @ cs) = (as @ (bs @ cs))
. . . . |     {append_assoc})
. . . . |    0
. . . . |
. . . . ⊢  (L↑n) = ((L @ (L↑n-1-1)) @ L)  by RWH (RevHypC 8) 0
. . . . |
. . . . ⊢  (L↑n) = ((L↑n-1) @ L)
. . . . | by RWH (AddrC [2] (RecUnfoldC ‘lpower‘)) 0
. . . . |
. . . . ⊢  if n=₂0→ nil else (L↑n-1) @ L fi = ((L↑n-1) @ L)
. . . .   by SplitOnConclITE
END
```