```
counter_lpower  THM

⊢  ∀e:ℤ, L:ℤ*, n:ℕ. ||e:(L↑n)|| = n·||e:L|| ∈ ℕ  by UnivCD
|
1. e : ℤ
2. L : ℤ*
3. n : ℕ
⊢  ||e:(L↑n)|| = n·||e:L||  by NatInd 3
|
|\
| .....basecase .....
| 2. L : ℤ*
| ⊢  ||e:(L↑0)|| = 0·||e:L||
| | by RW (RecUnfoldC ‘lpower‘) 0 THEN Reduce 0
| |
| ⊢  ||e:nil|| = 0·||e:L||
|   by RWH (LemmaC Thm* ∀e:ℤ. ||e:nil|| = 0 ∈ ℕ    {counter_of_nil})
|      0
|
.\
. .....upcase .....
. 3. n : ℤ
. 4. 0<n
. 5. ||e:(L↑n-1)|| = (n-1)·||e:L||
. ⊢   by RW (RecUnfoldC ‘lpower‘) 0
. |
. ⊢  ||e:if n=₂0→ nil else (L↑n-1) @ L fi|| = n·||e:L||
. | by SplitOnConclITE
. |
. |\
. | .....truecase .....
. | 6. n = 0
. | ⊢  ||e:nil|| = n·||e:L||  by Auto
. |
. .\
. . .....falsecase .....
. . 6. ¬n = 0
. . ⊢  ||e:(L↑n-1) @ L|| = n·||e:L||
. . | by RWH
. . |    (LemmaC
. . |     Thm* ∀e:ℤ, L,M:ℤ*. ||e:L @ M|| = ||e:L||+||e:M|| ∈ ℕ
. . |     {counter_append})
. . |    0
. . |
. . |\
. . | .....set predicate .....
. . | ⊢  0≤n·||e:L||  by Auto
. . |
. . .\
. . . ⊢  ||e:(L↑n-1)||+||e:L|| = n·||e:L||  by Auto
END
```