```
length_upto  THM

⊢  ∀n:ℕ. ||upto(n)|| ∼ n
| by InductionOnNat THEN Reduce 0 THEN Try Trivial
|
.....upcase .....
1. n : ℤ
2. 0<n
3. ||upto(n-1)|| ∼ (n-1)
⊢  ||upto(n)|| ∼ n  by RecUnfold ‘upto‘ 0 THEN SplitOnConclITE
|
|\
| .....truecase .....
| 1. n : ℤ
| 2. 0<n
| 3. ||upto(n-1)|| ∼ (n-1)
| 4. n = 0
| ⊢  ||nil|| ∼ n  by Auto
|
.\
. .....falsecase .....
. 1. n : ℤ
. 2. 0<n
. 3. ||upto(n-1)|| ∼ (n-1)
. 4. ¬n = 0
. ⊢  ||upto(n-1) @ [(n-1)]|| ∼ n
. | by Subst' (||upto(n-1) @ [(n-1)]|| = ||upto(n-1)||+1) 0
. |
. |\
. | 1. n : ℤ
. | 2. 0<n
. | 3. ||upto(n-1)|| ∼ (n-1)
. | 4. ¬n = 0
. | ⊢  ||upto(n-1) @ [(n-1)]|| = ||upto(n-1)||+1
. |   by RWO
. |      Thm* ∀as,bs:T List. ||as @ bs|| = ||as||+||bs|| ∈ ℤ
. |      {length_append}
. |      0
. |
. .\
. . 1. n : ℤ
. . 2. 0<n
. . 3. ||upto(n-1)|| ∼ (n-1)
. . 4. ¬n = 0
. . ⊢  (||upto(n-1)||+1) ∼ n  by Subst (||upto(n-1)|| ∼ (n-1)) 0
END
```