```
lang_sing_prod  THM

⊢  ∀L:LangOver(Alph). (Δ^0⊗L) = L
| by Unfold ‘languages‘ 0 THEN Unfold ‘lang_prod‘ 0
|    THEN
|    Unfold ‘lang_sing‘ 0
|    THEN
|    Unfold ‘lang_eq‘ 0
|    THEN
|    UnivCD
|
1. Alph : Type
2. L : Alph*→Prop
3. l : Alph*
⊢  (λl.∃i:{0...||l||}. (λl.null(l))(l[0..i⁻]) & L(l[i..||l||⁻]))(l)
|  ⟺
|  L(l)  by Reduce 0 THEN Analyze 0
|
|\
| ⊢  (∃i:{0...||l||}. null(l[0..i⁻]) & L(l[i..||l||⁻])) ⇒ L(l)
| | by Analyze 0
| |
| 4. ∃i:{0...||l||}. null(l[0..i⁻]) & L(l[i..||l||⁻])
| ⊢  L(l)  by Analyze 4
| |
| 4. i : {0...||l||}
| 5. null(l[0..i⁻]) & L(l[i..||l||⁻])
| ⊢   by Analyze 5
| |
| 5. null(l[0..i⁻])
| 6. L(l[i..||l||⁻])
| ⊢
| | by RWH
| |    (LemmaC Thm* ∀as:T*. null(as) ⟺ as = nil    {assert_of_null})
| |    5
| |
| 5. (l[0..i⁻]) = nil
| 6. L(l[i..||l||⁻])
| ⊢   by ApFunToHypEquands ‘x’ ||x|| ℤ 5
| |
| |\
| | .....fun wf .....
| | 7. x : Alph*
| | ⊢  ||x|| = ||x||  by Auto
| |
| .\
| . 7. ||l[0..i⁻]|| = ||nil||
| . ⊢
| . | by RWH
| . |    (LemmaC
| . |     Thm* ∀as:T*, i:{0...||as||}, j:{i...||as||}.
| . |          ||as[i..j⁻]|| = j-i
| . |     {length_segment})
| . |    7
| . |
```

```
| . 7. i-0 = ||nil||
| . ⊢   by Reduce 7
| . |
| . 7. i-0 = 0
| . ⊢   by Assert (i = 0)
| . |
| . |\
| . | .....assertion .....
| . | ⊢  i = 0  by Auto
| . |
| . .\
| . . 8. i = 0
| . . ⊢   by RWH (HypC 8) 6
| . . |
| . . 6. L(l[0..||l||⁻])
| . . 7. i-0 = 0
| . . 8. i = 0
| . . ⊢
| . .   by RWH
| . .      (LemmaC
| . .       Thm* ∀as:T*. (as[0..||as||⁻]) = as    {whole_segment})
| . .      6
|
.\
. ⊢  (∃i:{0...||l||}. null(l[0..i⁻]) & L(l[i..||l||⁻])) ⇐ (L(l))
. | by Analyze 0
. |
. 4. L(l)
. ⊢  ∃i:{0...||l||}. null(l[0..i⁻]) & L(l[i..||l||⁻])  by Witness 0
. |
. |\
. | ⊢  0≤||l||
. |   by Inst Thm* ∀l:A*. ||l||≥0    {non_neg_length} [Alph;l]
. |
. .\
. . ⊢  null(l[0..0⁻]) & L(l[0..||l||⁻])  by Analyze 0
. . |
. . |\
. . | ⊢  null(l[0..0⁻])
. . | | by RWH
. . | |    (LemmaC
. . | |     Thm* ∀as:T*. null(as) ⟺ as = nil    {assert_of_null})
. . | |    0
. . | |
. . | ⊢  (l[0..0⁻]) = nil  by Assert (||l[0..0⁻]|| = 0)
. . | |
. . | |\
. . | | .....assertion .....
. . | | ⊢  ||l[0..0⁻]|| = 0
. . | |   by RWH
. . | |      (LemmaC
. . | |       Thm* ∀as:T*, i:{0...||as||}, j:{i...||as||}.
. . | |            ||as[i..j⁻]|| = j-i
. . | |       {length_segment})
. . | |      0
```

```
. . | |      THEN
. . | |      Inst Thm* ∀l:A*. ||l||≥0    {non_neg_length} [Alph;l]
. . | |
. . | .\
. . | . 5. ||l[0..0⁻]|| = 0
. . | . ⊢   by Assert (∀m:Alph*. ||m|| = 0 ⇒ m = nil)
. . | . |
. . | . |\
. . | . | .....assertion .....
. . | . | ⊢  ∀m:Alph*. ||m|| = 0 ⇒ m = nil  by Analyze 0
. . | . | |
. . | . | 6. m : Alph*
. . | . | ⊢  ||m|| = 0 ⇒ m = nil  by ListInd 6
. . | . | |
. . | . | |\
. . | . | | ⊢  ||nil|| = 0 ⇒ nil = nil  by Analyze 0
. . | . | |
. . | . | .\
. . | . | . 7. u : Alph
. . | . | . 8. v : Alph*
. . | . | . 9. ||v|| = 0 ⇒ v = nil
. . | . | . ⊢  ||u.v|| = 0 ⇒ u.v = nil  by Analyze 0
. . | . | . |
. . | . | . 10. ||u.v|| = 0
. . | . | . ⊢  u.v = nil
. . | . | . | by RWH
. . | . | . |    (LemmaC
. . | . | . |     Thm* ∀a:A, as:A*. ||a.as|| = ||as||+1
. . | . | . |     {length_cons})
. . | . | . |    10
. . | . | . |
. . | . | . 10. ||v||+1 = 0
. . | . | . ⊢
. . | . | .   by Inst Thm* ∀l:A*. ||l||≥0    {non_neg_length}
. . | . | .      [Alph;v]
. . | . |
. . | . .\
. . | . . 6. ∀m:Alph*. ||m|| = 0 ⇒ m = nil
. . | . . ⊢   by Witness6 l[0..0⁻]
. . | . . |
. . | . . 6. ||l[0..0⁻]|| = 0 ⇒ (l[0..0⁻]) = nil
. . | . . ⊢   by Analyze 6
. . |
. . .\
. . . ⊢  L(l[0..||l||⁻])
. . .   by RWH
. . .      (LemmaC
. . .       Thm* ∀as:T*. (as[0..||as||⁻]) = as    {whole_segment})
. . .      0
END
```