```
lang_prod_sing  THM

⊢  ∀L:LangOver(Alph). (L⊗Δ⁰) = L
| by Unfold ‘lang_eq‘ 0 THEN Unfold ‘lang_prod‘ 0
|    THEN
|    Unfold ‘languages‘ 0
|    THEN
|    Unfold ‘lang_sing‘ 0
|    THEN
|    UnivCD
|
1. Alph : Type
2. L : Alph*→Prop
3. l : Alph*
⊢  (λl.∃i:{0...||l||}. L(l[0..i⁻]) & (λl.null(l))(l[i..||l||⁻]))(l)
|  ⟺
|  L(l)  by Reduce 0
|
⊢  (∃i:{0...||l||}. L(l[0..i⁻]) & null(l[i..||l||⁻])) ⟺ L(l)
| by Analyze 0 THEN Analyze 0
|
|\
| 4. ∃i:{0...||l||}. L(l[0..i⁻]) & null(l[i..||l||⁻])
| ⊢  L(l)  by Analyze 4
| |
| 4. i : {0...||l||}
| 5. L(l[0..i⁻]) & null(l[i..||l||⁻])
| ⊢   by Analyze 5
| |
| 5. L(l[0..i⁻])
| 6. null(l[i..||l||⁻])
| ⊢
| | by RWH
| |    (LemmaC Thm* ∀as:T*. null(as) ⟺ as = nil    {assert_of_null})
| |    6
| |
| 6. (l[i..||l||⁻]) = nil
| ⊢   by ApFunToHypEquands ‘x’ ||x|| ℤ 6
| |
| |\
| | .....fun wf .....
| | 7. x : Alph*
| | ⊢  ||x|| = ||x||  by Auto
| |
| .\
| . 7. ||l[i..||l||⁻]|| = ||nil||
| . ⊢
| . | by RWH
| . |    (LemmaC
| . |     Thm* ∀as:T*, i:{0...||as||}, j:{i...||as||}.
| . |           ||as[i..j⁻]|| = j-i
| . |     {length_segment})
| . |    7
| . |
| . 7. ||l||-i = ||nil||
```

```
| . ⊢   by Reduce 7
| . |
| . 7. ||l||-i = 0
| . ⊢   by Assert (||l|| = i)
| . |
| . |\
| . | .....assertion .....
| . | ⊢  ||l|| = i  by Auto
| . |
| . .\
| . . 8. ||l|| = i
| . . ⊢   by RWH (RevHypC 8) 5
| . . |
| . . 5. L(l[0..||l||⁻])
| . . 6. (l[i..||l||⁻]) = nil
| . . 7. ||l||-i = 0
| . . 8. ||l|| = i
| . . ⊢
| . .   by RWH
| . .      (LemmaC
| . .       Thm* ∀as:T*. (as[0..||as||⁻]) = as    {whole_segment})
| . .      5
|
.\
. 4. L(l)
. ⊢  ∃i:{0...||l||}. L(l[0..i⁻]) & null(l[i..||l||⁻])
. | by Witness ||l||
. |
. |\
. | ⊢  0≤||l||
. |   by Inst Thm* ∀l:A*. ||l||≥0    {non_neg_length} [Alph;l]
. |
. .\
. . ⊢  L(l[0..||l||⁻]) & null(l[||l||..||l||⁻])  by Analyze 0
. . |
. . |\
. . | ⊢  L(l[0..||l||⁻])
. . |   by RWH
. . |      (LemmaC
. . |       Thm* ∀as:T*. (as[0..||as||⁻]) = as    {whole_segment})
. . |      0
. . |
. . .\
. . . ⊢  null(l[||l||..||l||⁻])
. . . | by RWH
. . . |    (LemmaC
. . . |     Thm* ∀as:T*. null(as) ⟺ as = nil    {assert_of_null})
. . . |    0
. . . |
. . . ⊢  (l[||l||..||l||⁻]) = nil
. . . | by Assert (||l[||l||..||l||⁻]|| = 0)
. . . |
. . . |\
. . . | .....assertion .....
. . . | ⊢  ||l[||l||..||l||⁻]|| = 0
```

```
. . . | | by RWH
. . . | |    (LemmaC
. . . | |     Thm* ∀as:T*, i:{0...||as||}, j:{i...||as||}.
. . . | |          ||as[i..j⁻]|| = j-i
. . . | |     {length_segment})
. . . | |    0
. . . | |
. . . | |\
. . . | | ⊢  0≤||l||
. . . | |   by Inst Thm* ∀l:A*. ||l||≥0    {non_neg_length} [Alph;l]
. . . | |
. . . | .\
. . . | . ⊢  ||l||-||l|| = 0  by Auto
. . . |
. . . .\
. . . . 5. ||l[||l||..||l||⁻]|| = 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[||l||..||l||⁻]
. . . . . |
. . . . . 6. ||l[||l||..||l||⁻]|| = 0 ⇒ (l[||l||..||l||⁻]) = nil
. . . . . ⊢   by Analyze 6
END
```