```
lang_auto_compute  THM

⊢  ∀L:LangOver(Alph), g:((x,y:Alph*//(x L-induced Equiv y))→ℬ)
|  , l:Alph*. (Result(A(g))l) = l ∈ x,y:Alph*//(x L-induced Equiv y)
| by Analyze 0 THEN Analyze 0
|
1. Alph : Type
2. L : LangOver(Alph)
⊢  ∀g:((x,y:Alph*//(x L-induced Equiv y))→ℬ), l:Alph*.
|  (Result(A(g))l) = l
| by Inst
|    Thm* ∀L:LangOver(A). EquivRel x,y:A*. x L-induced Equiv y
|    {lang_rel_equi}
|    [Alph;L]
|
3. EquivRel x,y:Alph*. x L-induced Equiv y
⊢   by Assert (L ∈ Alph*→Prop)
|
|\
| .....assertion .....
| ⊢  L ∈ Alph*→Prop  by Fold ‘languages‘ 0
|
.\
. 4. L ∈ Alph*→Prop
. ⊢   by Auto
. |
. 5. g : (x,y:Alph*//(x L-induced Equiv y))→ℬ
. 6. l : Alph*
. ⊢  (Result(A(g))l) = l  by ListInd 6
. |
. |\
. | ⊢  (Result(A(g))nil) = nil
. | | by RecUnfold ‘compute_list‘ 0 THEN Reduce 0
. | |
. | ⊢  nil = nil  by Analyze
. | |
. | ⊢  nil L-induced Equiv nil
. | | by Inst
. | |    Thm* ∀L:LangOver(A). Refl(A*;x,y.x L-induced Equiv y)
. | |    {lang_rel_refl}
. | |    [Alph;L]
. | |
. | 7. Refl(Alph*;x,y.x L-induced Equiv y)
. | ⊢   by Unfold ‘refl‘ 7
. | |
. | 7. ∀a:Alph*. a L-induced Equiv a
. | ⊢   by BackThru 7
. |
. .\
. . 7. u : Alph
. . 8. v : Alph*
. . 9. (Result(A(g))v) = v
. . ⊢  (Result(A(g))u.v) = u.v  by RecCaseSplit ‘compute_list‘
. . |
. . |\
```

```
. . | 10. u.v = nil
. . | ⊢  nil = u.v  by ApFunToHypEquands ‘x’ ||x|| ℤ 10
. . | |
. . | 11. ||u.v|| = ||nil||
. . | ⊢   by Reduce 11
. . |
. . .\
. . . 10. ¬u.v = nil
. . . ⊢  u.(Result(A(g))v) = u.v  by HypSubst 9 0
. . . |
. . . |\
. . . | ⊢  u.v = u.v  by Auto
. . . |
. . . .\
. . . . 11. z : x,y:Alph*//(x L-induced Equiv y)
. . . . ⊢  u.z ∈ x,y:Alph*//(x L-induced Equiv y)  by Analyze 11
. . . . |
. . . . 11. z1 : Alph*
. . . . 12. z2 : Alph*
. . . . 13. z1 L-induced Equiv z2
. . . . ⊢  u.z1 = u.z2  by Analyze
. . . . |
. . . . ⊢  (u.z1) L-induced Equiv (u.z2)
. . . . | by Unfold ‘lang_rel‘ 0 THEN Reduce 0
. . . . |
. . . . ⊢  ∀z:Alph*. L(z @ (u.z1)) ⟺ L(z @ (u.z2))  by Analyze 0
. . . . |
. . . . 14. z : Alph*
. . . . ⊢  L(z @ (u.z1)) ⟺ L(z @ (u.z2))
. . . . | by Unfold ‘lang_rel‘ 13 THEN Reduce 13
. . . . |
. . . . 13. ∀z:Alph*. L(z @ z1) ⟺ L(z @ z2)
. . . . 14. z : Alph*
. . . . ⊢   by Witness13 z @ [u]
. . . . |
. . . . 13. z : Alph*
. . . . 14. L((z @ [u]) @ z1) ⟺ L((z @ [u]) @ z2)
. . . . ⊢
. . . . | by RWH
. . . . |    (LemmaC
. . . . |     Thm* ∀as,bs,cs:T*. ((as @ bs) @ cs) = (as @ (bs @ cs))
. . . . |     {append_assoc})
. . . . |    14
. . . . |
. . . . 14. L(z @ [u] @ z1) ⟺ L(z @ [u] @ z2)
. . . . ⊢   by Reduce 14
END
```