```
mn_31  THM

⊢  ∀L:LangOver(Alph).
|  Fin(Alph) ⇒
|  Fin(x,y:Alph*//L-induced Equiv(x,y)) & (∀l:Alph*. Dec(L(l))) ⇒
|  (∃St:Type, Auto:Automata(Alph;St). Fin(St) & L = LangOf(Auto))
| by Analyze 0
|
1. Alph : Type
⊢  ∀L:LangOver(Alph).
|  Fin(Alph) ⇒
|  Fin(x,y:Alph*//L-induced Equiv(x,y)) & (∀l:Alph*. Dec(L(l))) ⇒
|  (∃St:Type, Auto:Automata(Alph;St). Fin(St) & L = LangOf(Auto))
| by Analyze 0
|
2. L : LangOver(Alph)
⊢  Fin(Alph) ⇒
|  Fin(x,y:Alph*//L-induced Equiv(x,y)) & (∀l:Alph*. Dec(L(l))) ⇒
|  (∃St:Type, Auto:Automata(Alph;St). Fin(St) & L = LangOf(Auto))
| 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 UnivCD THENW (Auto THEN Fold ‘languages‘ 0)
|
4. Fin(Alph)
5. Fin(x,y:Alph*//L-induced Equiv(x,y)) & (∀l:Alph*. Dec(L(l)))
⊢  ∃St:Type, Auto:Automata(Alph;St). Fin(St) & L = LangOf(Auto)
| by Witness x,y:Alph*//L-induced Equiv(x,y)
|
⊢  ∃Auto:Automata(Alph;x,y:Alph*//L-induced Equiv(x,y)).
|  Fin(x,y:Alph*//L-induced Equiv(x,y)) & L = LangOf(Auto)
| by Analyze 5
|
5. Fin(x,y:Alph*//L-induced Equiv(x,y))
6. ∀l:Alph*. Dec(L(l))
⊢
| by Inst
|    Thm* ∀f:(T→Prop).
|         (∀t:T. Dec(f(t))) ⇒ (∃g:(T→𝔹). ∀t:T. f(t) ⟺ g(t))
|    {decid_is_comp}
|    [Alph*;L]
|    THENA
|    (Auto THEN Fold ‘languages‘ 0)
|
7. ∃g:(Alph*→𝔹). ∀t:Alph*. L(t) ⟺ g(t)
⊢   by Analyze 7
|
7. g : Alph*→𝔹
8. ∀t:Alph*. L(t) ⟺ g(t)
⊢
| by Assert
|    (<(λs,a. a.s),nil,g>
```

```
|     ∈ Automata(Alph;x,y:Alph*//L-induced Equiv(x,y)))
|
|\
| .....assertion .....
| ⊢  <(λs,a. a.s),nil,g>
| |  ∈ Automata(Alph;x,y:Alph*//L-induced Equiv(x,y))
| | by Unfold ‘automata‘ 0
| |
| ⊢  <(λs,a. a.s),nil,g>
| |  ∈ ((x,y:Alph*//L-induced Equiv(x,y))→Alph→
| |     (x,y:Alph*//L-induced Equiv(x,y)))
| |    ×(x,y:Alph*//L-induced Equiv(x,y))×((x,y:Alph*//L-induced Equiv
| |                                                        (x
| |                                                        ,y))
| |                                        →𝔹)  by Analyze
| |
| |\
| | ⊢  (λs,a. a.s)
| | |  ∈ (x,y:Alph*//L-induced Equiv(x,y))→Alph→
| | |    (x,y:Alph*//L-induced Equiv(x,y))  by Auto
| | |
| | 9. s : x,y:Alph*//L-induced Equiv(x,y)
| | 10. a : Alph
| | ⊢  a.s ∈ x,y:Alph*//L-induced Equiv(x,y)  by Analyze 9
| | |
| | 9. a : Alph
| | 10. s1 : Alph*
| | 11. s2 : Alph*
| | 12. L-induced Equiv(s1,s2)
| | ⊢  a.s1 = a.s2  by Analyze
| | |
| | ⊢  L-induced Equiv(a.s1,a.s2)
| | | by Unfold ‘lang_rel‘ 0 THEN Reduce 0 THEN Unfold ‘lang_rel‘ -1
| | |    THEN
| | |    Reduce -1
| | |
| | 12. ∀z:Alph*. L(z @ s1) ⟺ L(z @ s2)
| | ⊢  ∀z:Alph*. L(z @ (a.s1)) ⟺ L(z @ (a.s2))
| | | by (Analyze 0) THEN (InstHyp [z @ [a]] 12)
| | |
| | 13. z : Alph*
| | 14. L((z @ [a]) @ s1) ⟺ L((z @ [a]) @ s2)
| | ⊢  L(z @ (a.s1)) ⟺ L(z @ (a.s2))
| | | by Unfold ‘languages‘ 2
| | |    THEN
| | |    RWH
| | |    (LemmaC
| | |     Thm* ∀as,bs,cs:T*. ((as @ bs) @ cs) = (as @ (bs @ cs))
| | |     {append_assoc})
| | |    -1
| | |
| | 2. L : Alph*→Prop
| | 3. EquivRel x,y:Alph*. x L-induced Equiv y
| | 4. Fin(Alph)
| | 5. Fin(x,y:Alph*//L-induced Equiv(x,y))
```

```
| | 6. ∀l:Alph*. Dec(L(l))
| | 7. g : Alph*→𝔹
| | 8. ∀t:Alph*. L(t) ⟺ g(t)
| | 9. a : Alph
| | 10. s1 : Alph*
| | 11. s2 : Alph*
| | 12. ∀z:Alph*. L(z @ s1) ⟺ L(z @ s2)
| | 13. z : Alph*
| | 14. L(z @ [a] @ s1) ⟺ L(z @ [a] @ s2)
| | ⊢   by Reduce -1 THEN NthHyp -1
| |
| |\
| | ⊢  <nil,g>
| | |  ∈ (x,y:Alph*//L-induced Equiv(x,y))×
| | |    ((x,y:Alph*//L-induced Equiv(x,y))→𝔹)  by Analyze
| | |
| | |\
| | | ⊢  nil ∈ x,y:Alph*//L-induced Equiv(x,y)  by Analyze
| | |
| | |\
| | | ⊢  g ∈ (x,y:Alph*//L-induced Equiv(x,y))→𝔹
| | | | by ExtWith [] [Alph*→𝔹]
| | | |
| | | 9. x : x,y:Alph*//L-induced Equiv(x,y)
| | | ⊢  g(x) ∈ 𝔹  by Analyze -1
| | | |
| | | 9. x1 : Alph*
| | | 10. x2 : Alph*
| | | 11. L-induced Equiv(x1,x2)
| | | ⊢  g(x1) = g(x2)  by Unfold ‘lang_rel‘ -1 THEN Reduce -1
| | | |
| | | 11. ∀z:Alph*. L(z @ x1) ⟺ L(z @ x2)
| | | ⊢   by InstHyp [nil] -1
| | | |
| | | 12. L(nil @ x1) ⟺ L(nil @ x2)
| | | ⊢   by Reduce -1
| | | |
| | | 12. L(x1) ⟺ L(x2)
| | | ⊢
| | | | by BackThru
| | | |    Thm* ∀a,b:𝔹. (a ⟺ b) ⇒ a = b    {iff_imp_equal_bool}
| | | |
| | | ⊢  g(x1) ⟺ g(x2)  by (InstHyp [x1] 8) THEN (InstHyp [x2] 8)
| | | |
| | | 13. L(x1) ⟺ g(x1)
| | | 14. L(x2) ⟺ g(x2)
| | | ⊢
| | |   by (Unfold ‘languages‘ 2 THEN RWH (HypC -2) 12)
| | |      THEN
| | |      (RWH (HypC -1) 12)
| | |      THEN
| | |      (NthHyp 12)
| | |
| | .\
| | . .....eq aux .....
```

```
| | . 9. x,y:Alph*//L-induced Equiv(x,y)
| | . ⊢  (x,y:Alph*//L-induced Equiv(x,y))→𝔹 ∈ Type  by Auto
| |
| .\
| . .....eq aux .....
| . 9. (x,y:Alph*//L-induced Equiv(x,y))→Alph→
| .    (x,y:Alph*//L-induced Equiv(x,y))
| . ⊢  (x,y:Alph*//L-induced Equiv(x,y))×((x,y:Alph*//L-induced Equiv
| .                                                    (x
| .                                                    ,y))
| .                                          →𝔹)
| .    ∈ Type  by Auto
|
.\
. 9. <(λs,a. a.s),nil,g>
. |  ∈ Automata(Alph;x,y:Alph*//L-induced Equiv(x,y))
. ⊢   by Witness <(λs,a. a.s),nil,g>
. |
. ⊢  Fin(x,y:Alph*//L-induced Equiv(x,y))
. |  & L = LangOf(<(λs,a. a.s),nil,g>)  by Analyze 0
. |
. |\
. | ⊢  Fin(x,y:Alph*//L-induced Equiv(x,y))  by Auto
. |
. .\
. . ⊢  L = LangOf(<(λs,a. a.s),nil,g>)  by Unfold ‘lang_eq‘ 0
. . |
. . ⊢  ∀l:Alph*. L(l) ⟺ LangOf(<(λs,a. a.s),nil,g>)(l)  by Analyze 0
. . |
. . 10. l : Alph*
. . ⊢  L(l) ⟺ LangOf(<(λs,a. a.s),nil,g>)(l)
. . | by Unfold ‘auto_lang‘ 0
. . |
. . ⊢  L(l) ⟺ (λl.<(λs,a. a.s),nil,g>(l)↓)(l)  by Reduce 0
. . |
. . ⊢  L(l) ⟺ <(λs,a. a.s),nil,g>(l)↓
. . | by RWH (HypC 8) 0 THENW (Auto THEN Fold ‘languages‘ 0)
. . |
. . ⊢  g(l) ⟺ <(λs,a. a.s),nil,g>(l)↓
. . | by Assert (∀b1,b2:𝔹. b1 = b2 ⇒ (b1 ⟺ b2))
. . |
. . |\
. . | .....assertion .....
. . | ⊢  ∀b1,b2:𝔹. b1 = b2 ⇒ (b1 ⟺ b2)  by UnivCD
. . | |
. . | 11. b1 : 𝔹
. . | 12. b2 : 𝔹
. . | 13. b1 = b2
. . | ⊢  b1 ⟺ b2  by RWH (HypC 13) 0
. . |
. . .\
. . . 11. ∀b1,b2:𝔹. b1 = b2 ⇒ (b1 ⟺ b2)
. . . ⊢   by InstHyp [g(l);<(λs,a. a.s),nil,g>(l)↓] 11
. . . |
. . . .....antecedent .....
```

```
. . . ⊢  g(l) = <(λs,a. a.s),nil,g>(l)↓  by Unfold ‘accept_list‘ 0
. . . |
. . . ⊢  g(l)
. . . |  =
. . . |  FinalState(<(λs,a. a.s),nil,g>)(Result(<(λs,a. a.s),nil,g>)l)
. . . | by Unfold ‘DA_fin‘ 0
. . . |
. . . ⊢  g(l)
. . . |  =
. . . |  2of(2of(<(λs,a. a.s),nil,g>))(Result(<(λs,a. a.s),nil,g>)l)
. . . | by Reduce 0
. . . |
. . . ⊢  g(l) = g(Result(<(λs,a. a.s),nil,g>)l)  by Analyze
. . . |
. . . |\
. . . | ⊢  g = g  by Auto
. . . |
. . . .\
. . . . ⊢  l = (Result(<(λs,a. a.s),nil,g>)l)  by ListInd 10
. . . . |
. . . . |\
. . . . | ⊢  nil = (Result(<(λs,a. a.s),nil,g>)nil)
. . . . | | by RecCaseSplit ‘compute_list‘ THEN Reduce 12
. . . . | |
. . . . | 12. ¬True
. . . . | ⊢  nil = "?".(Result(<(λs,a. a.s),nil,g>)nil)
. . . . |   by Analyze 12
. . . . |
. . . . .\
. . . . . 12. u : Alph
. . . . . 13. v : Alph*
. . . . . 14. v = (Result(<(λs,a. a.s),nil,g>)v)
. . . . . ⊢  u.v = (Result(<(λs,a. a.s),nil,g>)u.v)
. . . . . | by RecCaseSplit ‘compute_list‘
. . . . . |
. . . . . |\
. . . . . | 15. u.v = nil
. . . . . | ⊢  u.v = nil  by Auto
. . . . . |
. . . . . .\
. . . . . . 15. ¬u.v = nil
. . . . . . ⊢  u.v = u.(Result(<(λs,a. a.s),nil,g>)v)
. . . . . .   by RWH (RevHypC 14) 0
END
```