```
mn_12  THM

⊢  ∀L:LangOver(Alph).
|  Fin(Alph) ⇒
|  (∃St:Type, Auto:Automata(Alph;St). Fin(St) & L = LangOf(Auto)) ⇒
|  (∃R:(Alph*→Alph*→Prop).
|   (EquivRel x,y:Alph*. x R y) c∧ (∃g:((x,y:Alph*//R(x,y))→𝔹).
|                                  Fin(x,y:Alph*//R(x,y))
|                                  & (∀l:Alph*. L(l) ⟺ g(l))
|                                  & (∀x,y,z:Alph*.
|                                     R(x,y) ⇒ R((z @ x),z @ y))))
| by UnivCD THEN Analyze 4 THEN Analyze 5 THEN Analyze 6
|
1. Alph : Type
2. L : LangOver(Alph)
3. Fin(Alph)
4. St : Type
5. Auto : Automata(Alph;St)
6. Fin(St)
7. L = LangOf(Auto)
⊢  ∃R:(Alph*→Alph*→Prop).
|  (EquivRel x,y:Alph*. x R y) c∧ (∃g:((x,y:Alph*//R(x,y))→𝔹).
|                                 Fin(x,y:Alph*//R(x,y))
|                                 & (∀l:Alph*. L(l) ⟺ g(l))
|                                 & (∀x,y,z:Alph*.
|                                    R(x,y) ⇒ R((z @ x),z @ y)))
| by Assert (EquivRel x,y:Alph*. (Result(Auto)x) = (Result(Auto)y))
|
|\
| .....assertion .....
| ⊢  EquivRel x,y:Alph*. (Result(Auto)x) = (Result(Auto)y)
|   by Analyze 0 THEN Analyze 0 THEN Try Auto THEN Analyze 0
|
.\
. 8. EquivRel x,y:Alph*. (Result(Auto)x) = (Result(Auto)y)
. ⊢   by Witness λx,y. (Result(Auto)x) = (Result(Auto)y)
. |
. |\
. | .....wf .....
. | ⊢  (λx,y. (Result(Auto)x) = (Result(Auto)y)) ∈ Alph*→Alph*→Prop
. |   by Auto
. |
. |\
. | ⊢  (EquivRel x,y:Alph*.
. | |            x (λx,y. (Result(Auto)x) = (Result(Auto)y)) y)
. | |  c∧ (∃g:((x,y:Alph*//(λx,y. (Result(Auto)x) = (Result(Auto)y))
. | |                      (x
. | |                      ,y))
. | |          →𝔹).
. | |
. | |     Fin(x,y:Alph*//(λx,y. (Result(Auto)x) = (Result(Auto)y))
. | |                    (x
. | |                    ,y))
. | |     & (∀l:Alph*. L(l) ⟺ g(l))
. | |     & (∀x,y,z:Alph*.
```

```
. | |        (λx,y. (Result(Auto)x) = (Result(Auto)y))(x,y) ⇒
. | |        (λx,y. (Result(Auto)x) = (Result(Auto)y))
. | |        ((z @ x)
. | |        ,z @ y)))  by Reduce 0
. | |
. | ⊢  (EquivRel x,y:Alph*. (Result(Auto)x) = (Result(Auto)y))
. | |  c∧ (∃g:((x,y:Alph*//((Result(Auto)x) = (Result(Auto)y)))→𝔹).
. | |     Fin(x,y:Alph*//((Result(Auto)x) = (Result(Auto)y)))
. | |     & (∀l:Alph*. L(l) ⟺ g(l))
. | |     & (∀x,y,z:Alph*.
. | |        (Result(Auto)x) = (Result(Auto)y) ⇒
. | |        (Result(Auto)z @ x) = (Result(Auto)z @ y)))
. | | by Analyze 0
. | |
. | |\
. | | ⊢  EquivRel x,y:Alph*. (Result(Auto)x) = (Result(Auto)y)
. | |   by Auto
. | |
. | .\
. | . 9. EquivRel x,y:Alph*. (Result(Auto)x) = (Result(Auto)y)
. | . ⊢  ∃g:((x,y:Alph*//((Result(Auto)x) = (Result(Auto)y)))→𝔹).
. | . |  Fin(x,y:Alph*//((Result(Auto)x) = (Result(Auto)y)))
. | . |  & (∀l:Alph*. L(l) ⟺ g(l))
. | . |  & (∀x,y,z:Alph*.
. | . |     (Result(Auto)x) = (Result(Auto)y) ⇒
. | . |     (Result(Auto)z @ x) = (Result(Auto)z @ y))  by Thin 9
. | . |
. | . 8. EquivRel x,y:Alph*. (Result(Auto)x) = (Result(Auto)y)
. | . ⊢   by Witness λl.Auto(l)↓
. | . |
. | . |\
. | . | .....wf .....
. | . | ⊢  (λl.Auto(l)↓)
. | . | |  ∈ (x,y:Alph*//((Result(Auto)x) = (Result(Auto)y)))→𝔹
. | . | | by Analyze
. | . | |
. | . | |\
. | . | | 9. l : x,y:Alph*//((Result(Auto)x) = (Result(Auto)y))
. | . | | ⊢  Auto(l)↓ ∈ 𝔹
. | . | |   by BackThru
. | . | |      Thm* ∀Auto:Automata(Alph;St)
. | . | |           , ...
. | . | |      {accept_list_qwf}
. | . | |
. | . | .\
. | . | . .....eq aux .....
. | . | . ⊢  x,y:Alph*//((Result(Auto)x) = (Result(Auto)y)) ∈ Type
. | . | .   by Auto
. | . | .
. | . |\
. | . | ⊢  Fin(x,y:Alph*//((Result(Auto)x) = (Result(Auto)y)))
. | . | |  & (∀l:Alph*. L(l) ⟺ (λl.Auto(l)↓)(l))
. | . | |  & (∀x,y,z:Alph*.
. | . | |     (Result(Auto)x) = (Result(Auto)y) ⇒
. | . | |     (Result(Auto)z @ x) = (Result(Auto)z @ y))
```

```
. | . | | by Analyze 0
. | . | |
. | . | |\
. | . | | ⊢  Fin(x,y:Alph*//((Result(Auto)x) = (Result(Auto)y)))
. | . | | | by Inst
. | . | | |    Thm* ∀f:(T→S).
. | . | | |         Fin(S) & (∀s:S. Dec(∃t:T. f(t) = s)) ⇒
. | . | | |         Fin(x,y:T//(f(x) = f(y)))
. | . | | |    {inv_of_fin_is_fin}
. | . | | |    [Alph*;St;λt.Result(Auto)t]
. | . | | |
. | . | | |\
. | . | | | .....antecedent .....
. | . | | | ⊢  Fin(St)
. | . | | | |  & (∀s:St. Dec(∃t:Alph*. (λt.Result(Auto)t)(t) = s))
. | . | | | | by Analyze 0
. | . | | | |
. | . | | | |\
. | . | | | | ⊢  Fin(St)  by Auto
. | . | | | |
. | . | | | .\
. | . | | | . ⊢  ∀s:St. Dec(∃t:Alph*. (λt.Result(Auto)t)(t) = s)
. | . | | | . | by Reduce 0
. | . | | | . |
. | . | | | . ⊢  ∀s:St. Dec(∃t:Alph*. (Result(Auto)t) = s)
. | . | | | . | by Analyze 0
. | . | | | . |
. | . | | | . 9. s : St
. | . | | | . ⊢  Dec(∃t:Alph*. (Result(Auto)t) = s)
. | . | | | .   by Inst
. | . | | | .      Thm* ∀Auto:Automata(Alph;St).
. | . | | | .           Fin(Alph) ⇒
. | . | | | .           Fin(St) ⇒
. | . | | | .           (∀s:St. Dec(∃w:Alph*. (Result(Auto)w) = s))
. | . | | | .      {reach_dec}
. | . | | | .      [Alph;St;Auto]
. | . | | |
. | . | | .\
. | . | | . 9. Fin(x,y:Alph*//((λt.Result(Auto)t)(x)
. | . | | .                   =
. | . | | .                   (λt.Result(Auto)t)(y)))
. | . | | . ⊢   by Reduce 9
. | . | |
. | . | .\
. | . | . ⊢  (∀l:Alph*. L(l) ⟺ (λl.Auto(l)↓)(l))
. | . | . |  & (∀x,y,z:Alph*.
. | . | . |     (Result(Auto)x) = (Result(Auto)y) ⇒
. | . | . |     (Result(Auto)z @ x) = (Result(Auto)z @ y))
. | . | . | by Analyze 0
. | . | . |
. | . | . |\
. | . | . | ⊢  ∀l:Alph*. L(l) ⟺ (λl.Auto(l)↓)(l)
. | . | . | | by RWH (HypC 7) 0 THEN Unfold ‘languages‘ 2
. | . | . | |    THEN
. | . | . | |    Try (Reduce 0)
```

```
. | . | . | |    THEN
. | . | . | |    Try Auto
. | . | . | |    THEN
. | . | . | |    Try (Fold ‘languages‘ 0)
. | . | . | |
. | . | . | |\
. | . | . | | 2. L : Alph*→Prop
. | . | . | | 3. Fin(Alph)
. | . | . | | 4. St : Type
. | . | . | | 5. Auto : Automata(Alph;St)
. | . | . | | 6. Fin(St)
. | . | . | | 7. L = LangOf(Auto)
. | . | . | | 8. EquivRel x,y:Alph*. (Result(Auto)x) = (Result(Auto)y)
. | . | . | | 9. Alph*
. | . | . | | ⊢  L ∈ LangOver(Alph)  by Fold ‘languages‘ 2
. | . | . | |
. | . | . | |\
. | . | . | | 2. L : Alph*→Prop
. | . | . | | 3. Fin(Alph)
. | . | . | | 4. St : Type
. | . | . | | 5. Auto : Automata(Alph;St)
. | . | . | | 6. Fin(St)
. | . | . | | 7. L = LangOf(Auto)
. | . | . | | 8. EquivRel x,y:Alph*. (Result(Auto)x) = (Result(Auto)y)
. | . | . | | 9. l : Alph*
. | . | . | | 10. LangOf(Auto)(l)
. | . | . | | ⊢  Auto(l)↓  by Unfold ‘auto_lang‘ 10 THEN Reduce 10
. | . | . | |
. | . | . | .\
. | . | . | . 2. L : Alph*→Prop
. | . | . | . 3. Fin(Alph)
. | . | . | . 4. St : Type
. | . | . | . 5. Auto : Automata(Alph;St)
. | . | . | . 6. Fin(St)
. | . | . | . 7. L = LangOf(Auto)
. | . | . | . 8. EquivRel x,y:Alph*. (Result(Auto)x) = (Result(Auto)y)
. | . | . | . 9. l : Alph*
. | . | . | . 10. Auto(l)↓
. | . | . | . ⊢  LangOf(Auto)(l)
. | . | . | .   by Unfold ‘auto_lang‘ 0 THEN Reduce 0
. | . | . | |
. | . | . | .\
. | . | . | . ⊢  ∀x,y,z:Alph*.
. | . | . | .  |  (Result(Auto)x) = (Result(Auto)y) ⇒
. | . | . | .  |  (Result(Auto)z @ x) = (Result(Auto)z @ y)  by UnivCD
. | . | . | .  |
. | . | . | . 9. x : Alph*
. | . | . | . 10. y : Alph*
. | . | . | . 11. z : Alph*
. | . | . | . 12. (Result(Auto)x) = (Result(Auto)y)
. | . | . | . ⊢  (Result(Auto)z @ x) = (Result(Auto)z @ y)
. | . | . | .   by BackThru
. | . | . | .      Thm* ∀Auto:Automata(Alph;St), x,y,z:Alph*.
. | . | . | .           (Result(Auto)x) = (Result(Auto)y) ⇒
. | . | . | .           (Result(Auto)z @ x) = (Result(Auto)z @ y)
```

```
. | . | . .      {compute_1_inv}
. | . |
. | . .\
. | . . .....wf .....
. | . . 9. g : (x,y:Alph*//((Result(Auto)x) = (Result(Auto)y)))→𝔹
. | . . ⊢  (Fin(x,y:Alph*//((Result(Auto)x) = (Result(Auto)y)))
. | . .     & (∀l:Alph*. L(l) ⟺ g(l))
. | . .     & (∀x,y,z:Alph*.
. | . .        (Result(Auto)x) = (Result(Auto)y) ⇒
. | . .        (Result(Auto)z @ x) = (Result(Auto)z @ y)))
. | . .    ∈ Prop  by Unfold ‘languages‘ 2
. |
. .\
. . .....wf .....
. . 9. R : Alph*→Alph*→Prop
. . ⊢  (EquivRel x,y:Alph*. x R y) c∧ (∃g:((x,y:Alph*//R(x,y))→𝔹).
. .                                    Fin(x,y:Alph*//R(x,y))
. .                                    & (∀l:Alph*. L(l) ⟺ g(l))
. .                                    & (∀x,y,z:Alph*.
. .                                       R(x,y) ⇒ R((z @ x),z @ y)))
. .    ∈ Prop  by Unfold ‘languages‘ 2
END
```