```
mn_13  THM

⊢  ∀Auto:Automata(Alph;St).
|  Fin(Alph) & Fin(St) ⇒
|  Fin(x,y:Alph*//(x LangOf(Auto)-induced Equiv y))  by UnivCD
|
1. Alph : Type
2. St : Type
3. Auto : Automata(Alph;St)
4. Fin(Alph) & Fin(St)
⊢  Fin(x,y:Alph*//(x LangOf(Auto)-induced Equiv y))
| by Inst
|    Thm* ∀L:LangOver(A). EquivRel x,y:A*. x L-induced Equiv y
|    {lang_rel_equi}
|    [Alph;LangOf(Auto)]
|
5. EquivRel x,y:Alph*. x LangOf(Auto)-induced Equiv y
⊢
| by Inst
|    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))))
|    {mn_12}
|    [Alph;LangOf(Auto)]
|
|\
| 4. Fin(Alph)
| 5. Fin(St)
| 6. EquivRel x,y:Alph*. x LangOf(Auto)-induced Equiv y
| ⊢  ∃St:Type, Auto@0:Automata(Alph;St).
| |  Fin(St) & LangOf(Auto) = LangOf(Auto@0)  by InstConcl [St;Auto]
| |
| ⊢  Fin(St) & LangOf(Auto) = LangOf(Auto)  by Analyze 0
| |
| |\
| | ⊢  Fin(St)  by Auto
| |
| .\
| . ⊢  LangOf(Auto) = LangOf(Auto)
| .   by Unfold ‘lang_eq‘ 0 THEN Fold ‘languages‘ 0
|
.\
. 6. ∃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*.
```

```
. |                                    LangOf(Auto)(l) ⟺ g(l))
. |                                 & (∀x,y,z:Alph*.
. |                                    R(x,y) ⇒ R((z @ x),z @ y)))
. ⊢   by Analyze 6
. |
. 6. R : Alph*→Alph*→Prop
. 7. (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*.
. |                                    LangOf(Auto)(l) ⟺ g(l))
. |                                 & (∀x,y,z:Alph*.
. |                                    R(x,y) ⇒ R((z @ x),z @ y)))
. ⊢   by Analyze 7
. |
. 7. EquivRel x,y:Alph*. x R y
. 8. ∃g:((x,y:Alph*//R(x,y))→𝔹).
. |  Fin(x,y:Alph*//R(x,y))
. |  & (∀l:Alph*. LangOf(Auto)(l) ⟺ g(l))
. |  & (∀x,y,z:Alph*. R(x,y) ⇒ R((z @ x),z @ y))
. ⊢   by Analyze 8
. |
. 8. g : (x,y:Alph*//R(x,y))→𝔹
. 9. Fin(x,y:Alph*//R(x,y))
. |  & (∀l:Alph*. LangOf(Auto)(l) ⟺ g(l))
. |  & (∀x,y,z:Alph*. R(x,y) ⇒ R((z @ x),z @ y))
. ⊢   by Analyze 9
. |
. 9. Fin(x,y:Alph*//R(x,y))
. 10. (∀l:Alph*. LangOf(Auto)(l) ⟺ g(l))
. |   & (∀x,y,z:Alph*. R(x,y) ⇒ R((z @ x),z @ y))
. ⊢   by Analyze 10
. |
. 10. ∀l:Alph*. LangOf(Auto)(l) ⟺ g(l)
. 11. ∀x,y,z:Alph*. R(x,y) ⇒ R((z @ x),z @ y)
. ⊢
. | by RWH
. |    (LemmaC Thm* Fin(T) ⟺ (∃m:ℕ. ℕm ∼ T)    {fin_iff_1_1_nsub})
. |    9
. |
. 9. ∃m:ℕ. ℕm ∼ (x,y:Alph*//R(x,y))
. 10. ∀l:Alph*. LangOf(Auto)(l) ⟺ g(l)
. 11. ∀x,y,z:Alph*. R(x,y) ⇒ R((z @ x),z @ y)
. ⊢   by Analyze 9
. |
. 9. m : ℕ
. 10. ℕm ∼ (x,y:Alph*//R(x,y))
. 11. ∀l:Alph*. LangOf(Auto)(l) ⟺ g(l)
. 12. ∀x,y,z:Alph*. R(x,y) ⇒ R((z @ x),z @ y)
. ⊢
. | by Inst
. |    Thm* ∀n:{1...}, A:Type, L:LangOver(A), R:(A*→A*→Prop).
. |         Fin(A) ⇒
. |         (EquivRel x,y:A*. x R y) ⇒
. |         (ℕn ∼ (x,y:A*//(x R y))) ⇒
. |         (∀x,y,z:A*. (x R y) ⇒ ((z @ x) R (z @ y))) ⇒
```

```
. |          (∃g:((x,y:A*//(x R y))→𝔹). ∀l:A*. L(l) ⟺ g(l)) ⇒
. |          (∃m:ℕ. ℕm ∼ (x,y:A*//(x L-induced Equiv y)))
. |          & (∀l:A*. Dec(L(l)))
. |     {mn_23}
. |     [m;Alph;LangOf(Auto);R]
. |
. |\
. | .....set predicate .....
. | 4. Fin(Alph)
. | 5. Fin(St)
. | 6. EquivRel x,y:Alph*. x LangOf(Auto)-induced Equiv y
. | 7. R : Alph*→Alph*→Prop
. | 8. EquivRel x,y:Alph*. x R y
. | 9. g : (x,y:Alph*//R(x,y))→𝔹
. | 10. m : ℕ
. | 11. ℕm ∼ (x,y:Alph*//R(x,y))
. | 12. ∀l:Alph*. LangOf(Auto)(l) ⟺ g(l)
. | 13. ∀x,y,z:Alph*. R(x,y) ⇒ R((z @ x),z @ y)
. | ⊢  1≤m  by Analyze 11
. | |
. | 11. f : ℕm→(x,y:Alph*//R(x,y))
. | 12. ∃g:((x,y:Alph*//R(x,y))→ℕm).
. | |   InvFuns(ℕm; x,y:Alph*//R(x,y); f; g)
. | 13. ∀l:Alph*. LangOf(Auto)(l) ⟺ g(l)
. | 14. ∀x,y,z:Alph*. R(x,y) ⇒ R((z @ x),z @ y)
. | ⊢   by Analyze 12
. | |
. | 12. g1 : (x,y:Alph*//R(x,y))→ℕm
. | 13. InvFuns(ℕm; x,y:Alph*//R(x,y); f; g1)
. | 14. ∀l:Alph*. LangOf(Auto)(l) ⟺ g(l)
. | 15. ∀x,y,z:Alph*. R(x,y) ⇒ R((z @ x),z @ y)
. | ⊢   by Assert (g1(nil) ∈ ℕm)
. | |
. | |\
. | | ⊢  nil ∈ x,y:Alph*//R(x,y)  by Analyze
. | |
. | .\
. | . 16. g1(nil) ∈ ℕm
. | . ⊢   by Analyze 16
. |
. |\
. | 4. Fin(Alph)
. | 5. Fin(St)
. | 6. EquivRel x,y:Alph*. x LangOf(Auto)-induced Equiv y
. | 7. R : Alph*→Alph*→Prop
. | 8. EquivRel x,y:Alph*. x R y
. | 9. g : (x,y:Alph*//R(x,y))→𝔹
. | 10. m : ℕ
. | 11. ℕm ∼ (x,y:Alph*//R(x,y))
. | 12. ∀l:Alph*. LangOf(Auto)(l) ⟺ g(l)
. | 13. ∀x,y,z:Alph*. R(x,y) ⇒ R((z @ x),z @ y)
. | ⊢  ∃g:((x,y:Alph*//(x R y))→𝔹). ∀l:Alph*. LangOf(Auto)(l) ⟺ g(l)
. |   by Witness g THEN Fold ‘languages‘ 0
. |
. .\
```

```
. . 13. (∃m:ℕ. ℕm ∼ (x,y:Alph*//(x LangOf(Auto)-induced Equiv y)))
. .     & (∀l:Alph*. Dec(LangOf(Auto)(l)))
. . ⊢
. .   by RWH
. .      (RevLemmaC
. .       Thm* Fin(T) ⟺ (∃m:ℕ. ℕm ∼ T)    {fin_iff_1_1_nsub})
. .      13
. .      THENW
. .      (Auto THEN Fold ‘languages‘ 0)
END
```