```
min_is_unique  THM

⊢  ∀Auto:Automata(Alph;St).
|  Con(Auto)
|  & (St ∼ (x,y:Alph*//(x LangOf(Auto)-induced Equiv y)))
|  & Fin(Alph)
|  & Fin(St)
|  ⇒
|  Auto ≡ A(λl.Auto(l)↓)  by Analyze 0 THEN Analyze 0 THEN Analyze 0
|
1. Alph : Type
2. St : Type
3. Auto : Automata(Alph;St)
⊢  Con(Auto)
|  & (St ∼ (x,y:Alph*//(x LangOf(Auto)-induced Equiv y)))
|  & Fin(Alph)
|  & Fin(St)
|  ⇒
|  Auto ≡ A(λl.Auto(l)↓)
| by Inst
|    Thm* ∀L:LangOver(A). EquivRel x,y:A*. x L-induced Equiv y
|    {lang_rel_equi}
|    [Alph;LangOf(Auto)]
|
4. EquivRel x,y:Alph*. x LangOf(Auto)-induced Equiv y
⊢   by Auto
|
5. Con(Auto)
6. St ∼ (x,y:Alph*//(x LangOf(Auto)-induced Equiv y))
7. Fin(Alph)
8. Fin(St)
⊢  Auto ≡ A(λl.Auto(l)↓)
| by Unfold ‘connected‘ 5
|    THEN
|    FwdThru
|    Thm* ∀Q:(A→B→Prop).
|         (∀x:A. ∃y:B. Q(x,y)) ⇒ (∃f:(A→B). ∀x:A. Q(x,f(x)))
|    {ax_choice}
|    [5]
|    THEN
|    Thin 5
|    THEN
|    Analyze 8
|
5. St ∼ (x,y:Alph*//(x LangOf(Auto)-induced Equiv y))
6. Fin(Alph)
7. Fin(St)
8. f : St→Alph*
9. ∀s:St. (Result(Auto)f(s)) = s
⊢   by Unfold ‘auto_iso‘ 0
|
⊢  ∃f:(St→(x,y:Alph*//(x LangOf(Auto)-induced Equiv y))).
|  Bij(St; x,y:Alph*//(x LangOf(Auto)-induced Equiv y); f)
|  & (∀s:St, a:Alph. f(δAuto(s,a)) = δA(λl.Auto(l)↓)(f(s),a))
|  & f(InitialState(Auto)) = InitialState(A(λl.Auto(l)↓))
```

```
|  & (∀s:St. FinalState(Auto)(s) = FinalState(A(λl.Auto(l)↓))(f(s)))
| by Witness f THEN Reduce 0
|
|\
| ⊢  Bij(St; x,y:Alph*//(x LangOf(Auto)-induced Equiv y); f)
| |  & (∀s:St, a:Alph. f(δAuto(s,a)) = a.f(s))
| |  & f(InitialState(Auto)) = nil
| |  & (∀s:St. FinalState(Auto)(s) = Auto(f(s))↓)  by Auto
| |
| |\
| | ⊢  Bij(St; x,y:Alph*//(x LangOf(Auto)-induced Equiv y); f)
| | | by Analyze 0
| | |
| | |\
| | | ⊢  Inj(St; x,y:Alph*//(x LangOf(Auto)-induced Equiv y); f)
| | | | by BackThru
| | | |    Thm* ∀Auto:Automata(Alph;St), c:(St→Alph*).
| | | |          (∀q:St. (Result(Auto)c(q)) = q)
| | | |          & Fin(Alph)
| | | |          & Fin(St)
| | | |          & (St ∼ (x,y:Alph*//(x LangOf(Auto)-induced Equiv y)))
| | | |          ⇒
| | | |
| | | |          Inj(St; x,y:Alph*//(x LangOf(Auto)-induced Equiv y);
| | | |              c)
| | | |    {homo_is_inj}
| | | |
| | | 10. q : St
| | | ⊢  (Result(Auto)f(q)) = q  by BackThru 9
| | |
| | .\
| | . ⊢  Surj(St; x,y:Alph*//(x LangOf(Auto)-induced Equiv y); f)
| | . | by BackThru
| | . |    Thm* ∀Auto:Automata(Alph;St), c:(St→Alph*).
| | . |          (∀q:St. (Result(Auto)c(q)) = q) & Fin(Alph) & Fin(St)
| | . |          ⇒
| | . |
| | . |          Surj(St; x,y:Alph*//(x LangOf(Auto)-induced Equiv y);
| | . |               c)
| | . |    {homo_is_surj}
| | . |
| | . 10. q : St
| | . ⊢  (Result(Auto)f(q)) = q  by BackThru 9
| |
| |\
| | 10. s : St
| | 11. a : Alph
| | ⊢  f(δAuto(s,a)) = a.f(s)
| |   by Inst
| |      Thm* ∀Auto:Automata(Alph;St)
| |           , g:((x,y:Alph*//(x LangOf(Auto)-induced Equiv y))→𝔹)
| |           , c:(St→Alph*).
| |           (∀q:St. (Result(Auto)c(q)) = q) ⇒
| |           (∀q:St, a:Alph.
| |            c(δAuto(q,a))
```

```
| |             =
| |             δA(g)(c(q),a)
| |             ∈ x,y:Alph*//(x LangOf(Auto)-induced Equiv y))
| |      {homo_step}
| |      [Alph;St;Auto;λt.true₂;f;s;a]
| |      THEN
| |      Reduce 12
| |
| |\
| | ⊢  f(InitialState(Auto)) = nil
| |   by Inst
| |      Thm* ∀Auto:Automata(Alph;St)
| |           , g:((x,y:Alph*//(x LangOf(Auto)-induced Equiv y))→𝔹)
| |           , c:(St→Alph*).
| |           (∀q:St. (Result(Auto)c(q)) = q) ⇒
| |           c(InitialState(Auto))
| |           =
| |           nil
| |           ∈ x,y:Alph*//(x LangOf(Auto)-induced Equiv y)
| |      {homo_init}
| |      [Alph;St;Auto;λt.true₂;f]
| |
| .\
| . 10. s : St
| . ⊢  FinalState(Auto)(s) = Auto(f(s))↓
| . | by Unfold `accept_list` 0 THEN Reduce 0
| . |
| . ⊢  FinalState(Auto)(s) = FinalState(Auto)(Result(Auto)f(s))
| .   by RWH (HypC 9) 0
|
|\
| 10. f1 : St→(x,y:Alph*//(x LangOf(Auto)-induced Equiv y))
| 11. s : St
| 12. a : Alph
| ⊢  a.f1(s) ∈ x,y:Alph*//(x LangOf(Auto)-induced Equiv y)
| | by Assert
| |    (∀q:x,y:Alph*//(x LangOf(Auto)-induced Equiv y), b:Alph.
| |     b.q ∈ x,y:Alph*//(x LangOf(Auto)-induced Equiv y))
| |
| |\
| | .....assertion .....
| | ⊢  ∀q:x,y:Alph*//(x LangOf(Auto)-induced Equiv y), b:Alph.
| | |  b.q ∈ x,y:Alph*//(x LangOf(Auto)-induced Equiv y)  by UnivCD
| | |
| | 13. q : x,y:Alph*//(x LangOf(Auto)-induced Equiv y)
| | 14. b : Alph
| | ⊢  b.q ∈ x,y:Alph*//(x LangOf(Auto)-induced Equiv y)
| | | by Analyze 13
| | |
| | 13. b : Alph
| | 14. q1 : Alph*
| | 15. q2 : Alph*
| | 16. q1 LangOf(Auto)-induced Equiv q2
| | ⊢  b.q1 = b.q2  by Unfold `lang_rel` 16 THEN Reduce 16
| | |
```

```
| | 16. ∀z:Alph*. LangOf(Auto)(z @ q1) ⟺ LangOf(Auto)(z @ q2)
| | ⊢   by Analyze
| | |
| | ⊢  (b.q1) LangOf(Auto)-induced Equiv (b.q2)
| | | by Unfold ‘lang_rel‘ 0 THEN Reduce 0
| | |
| | ⊢  ∀z:Alph*. LangOf(Auto)(z @ (b.q1)) ⟺ LangOf(Auto)(z @ (b.q2))
| | | by Analyze 0
| | |
| | 17. z : Alph*
| | ⊢  LangOf(Auto)(z @ (b.q1)) ⟺ LangOf(Auto)(z @ (b.q2))
| | | by Witness16 z @ [b]
| | |
| | 16. z : Alph*
| | 17. LangOf(Auto)((z @ [b]) @ q1) ⟺ LangOf(Auto)((z @ [b]) @ q2)
| | ⊢
| |   by RWH
| |      (LemmaC
| |       Thm* ∀as,bs,cs:T*. ((as @ bs) @ cs) = (as @ (bs @ cs))
| |       {append_assoc})
| |      17
| |      THENL
| |      [Fold ‘languages‘ 0;Fold ‘languages‘ 0;Reduce 17]
| |
| .\
| . 13. ∀q:x,y:Alph*//(x LangOf(Auto)-induced Equiv y), b:Alph.
| .     b.q ∈ x,y:Alph*//(x LangOf(Auto)-induced Equiv y)
| . ⊢   by BackThru 13
|
|\
| 10. St→(x,y:Alph*//(x LangOf(Auto)-induced Equiv y))
| ⊢  nil ∈ x,y:Alph*//(x LangOf(Auto)-induced Equiv y)  by Analyze
|
.\
. 10. f1 : St→(x,y:Alph*//(x LangOf(Auto)-induced Equiv y))
. 11. s : St
. ⊢  Auto(f1(s))↓ ∈ 𝔹
. | by Assert
. |    (∀q:x,y:Alph*//(x LangOf(Auto)-induced Equiv y). Auto(q)↓ ∈ 𝔹)
. |
. |\
. | .....assertion .....
. | ⊢  ∀q:x,y:Alph*//(x LangOf(Auto)-induced Equiv y). Auto(q)↓ ∈ 𝔹
. | | by Analyze 0
. | |
. | 12. q : x,y:Alph*//(x LangOf(Auto)-induced Equiv y)
. | ⊢  Auto(q)↓ ∈ 𝔹  by Analyze 12
. | |
. | 12. q1 : Alph*
. | 13. q2 : Alph*
. | 14. q1 LangOf(Auto)-induced Equiv q2
. | ⊢  Auto(q1)↓ = Auto(q2)↓  by Unfold ‘lang_rel‘ 14 THEN Reduce 14
. | |
. | 14. ∀z:Alph*. LangOf(Auto)(z @ q1) ⟺ LangOf(Auto)(z @ q2)
. | ⊢   by Witness14 nil THEN Reduce 14
```

```
. | |
. | 14. LangOf(Auto)(q1) ⟺ LangOf(Auto)(q2)
. | ⊢   by Unfold ‘auto_lang‘ 14 THEN Reduce 14
. | |
. | 14. Auto(q1)↓ ⟺ Auto(q2)↓
. | ⊢
. |   by BackThru
. |      Thm* ∀a,b:𝔹. (a ⟺ b) ⇒ a = b    {iff_imp_equal_bool}
. |
. .\
. . 12. ∀q:x,y:Alph*//(x LangOf(Auto)-induced Equiv y). Auto(q)↓ ∈ 𝔹
. . ⊢   by BackThru 12
END
```