```
any_iso_min_auto  THM

⊢  ∀Auto:Automata(Alph;St), S:Type, A:Automata(Alph;S).
|  Fin(Alph) ⇒
|  Fin(S) ⇒
|  Con(A) ⇒
|  (S ∼ (x,y:Alph*//(x LangOf(Auto)-induced Equiv y))) ⇒
|  LangOf(Auto) = LangOf(A) ⇒ A ≡ MinAuto(Auto)  by UnivCD
|
|\
| 1. Alph : Type
| 2. St : Type
| 3. Auto : Automata(Alph;St)
| 4. S : Type
| 5. A : Automata(Alph;S)
| 6. Fin(Alph)
| 7. Fin(S)
| 8. Con(A)
| 9. S ∼ (x,y:Alph*//(x LangOf(Auto)-induced Equiv y))
| 10. LangOf(Auto) = LangOf(A)
| ⊢  A ≡ MinAuto(Auto)
| | by Inst
| |    Thm* ∀L:LangOver(A). EquivRel x,y:A*. x L-induced Equiv y
| |    {lang_rel_equi}
| |    [Alph;LangOf(Auto)]
| |    THEN
| |    Inst
| |    Thm* ∀L:LangOver(A). EquivRel x,y:A*. x L-induced Equiv y
| |    {lang_rel_equi}
| |    [Alph;LangOf(A)]
| |
| 11. EquivRel x,y:Alph*. x LangOf(Auto)-induced Equiv y
| 12. EquivRel x,y:Alph*. x LangOf(A)-induced Equiv y
| ⊢
| | by Inst
| |    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)↓)
| |    {min_is_unique}
| |    [Alph;S;A]
| |
| |\
| | ⊢  S ∼ (x,y:Alph*//(x LangOf(A)-induced Equiv y))
| | | by Assert
| | |    (x,y:Alph*//(x LangOf(Auto)-induced Equiv y)
| | |     =
| | |     x,y:Alph*//(x LangOf(A)-induced Equiv y))
| | |
| | |\
| | | .....assertion .....
| | | ⊢  x,y:Alph*//(x LangOf(Auto)-induced Equiv y)
```

```
| | |    =
| | |    x,y:Alph*//(x LangOf(A)-induced Equiv y)
| | |   by BackThru
| | |      Thm* ∀A1:Automata(Alph;S1), A2:Automata(Alph;S2).
| | |           LangOf(A1) = LangOf(A2) ⇒
| | |           x,y:Alph*//(x LangOf(A1)-induced Equiv y)
| | |           =
| | |           x,y:Alph*//(x LangOf(A2)-induced Equiv y)
| | |      {lang_eq_imp_quo_eq}
| | |
| | .\
| | . 13. x,y:Alph*//(x LangOf(Auto)-induced Equiv y)
| | .     =
| | .     x,y:Alph*//(x LangOf(A)-induced Equiv y)
| | . ⊢   by RWH (HypC 13) 9
| |
| .\
| . 13. A ≡ A(λl.A(l)↓)
| . ⊢   by Unfold `min_auto` 0
| . |
| . ⊢  A ≡ A(λl.Auto(l)↓)  by Assert (MinAuto(Auto) = MinAuto(A))
| . |
| . |\
| . | .....assertion .....
| . | ⊢  MinAuto(Auto) = MinAuto(A)  by Unfold `min_auto` 0
| . | |
| . | ⊢  A(λl.Auto(l)↓) = A(λl.A(l)↓)  by Unfold `lang_auto` 0
| . | |
| . | ⊢  <(λs,a. a.s),nil,(λl.Auto(l)↓)>
| . | |  =
| . | |  <(λs,a. a.s),nil,(λl.A(l)↓)>
| . | | by Unfold `automata` 0 THEN Analyze THENL [Auto;Analyze]
| . | |    THENL
| . | |    [Auto;Auto;Analyze]
| . | |
| . | |\
| . | | 14. s : x,y:Alph*//(x LangOf(Auto)-induced Equiv y)
| . | | 15. a : Alph
| . | | ⊢  a.s = a.s  by Analyze 14
| . | | |
| . | | 14. a : Alph
| . | | 15. s1 : Alph*
| . | | 16. s2 : Alph*
| . | | 17. s1 LangOf(Auto)-induced Equiv s2
| . | | ⊢  a.s1 = a.s2  by Analyze
| . | | |
| . | | ⊢  (a.s1) LangOf(Auto)-induced Equiv (a.s2)
| . | | | by Unfold `lang_rel` 17 THEN Reduce 17
| . | | |
| . | | 17. ∀z:Alph*. LangOf(Auto)(z @ s1) ⟺ LangOf(Auto)(z @ s2)
| . | | ⊢   by Unfold `lang_rel` 0 THEN Reduce 0 THEN Analyze 0
| . | | |
| . | | 18. z : Alph*
| . | | ⊢  LangOf(Auto)(z @ (a.s1)) ⟺ LangOf(Auto)(z @ (a.s2))
| . | | | by Witness17 z @ [a] THEN Reduce 18
```

```
| . | | |
| . | | 17. z : Alph*
| . | | 18. LangOf(Auto)((z @ [a]) @ s1)
| . | |     ⟺
| . | |     LangOf(Auto)((z @ [a]) @ s2)
| . | | ⊢
| . | |   by RWH
| . | |      (LemmaC
| . | |       Thm* ∀as,bs,cs:T*. ((as @ bs) @ cs) = (as @ (bs @ cs))
| . | |       {append_assoc})
| . | |      18
| . | |      THEN
| . | |      Try (Fold ‘languages‘ 0)
| . | |      THEN
| . | |      Reduce 18
| . | |
| . | |\
| . | | ⊢  nil = nil  by Auto
| . | | |
| . | | ⊢   by Analyze
| . | |
| . | |\
| . | | 14. l : x,y:Alph*//(x LangOf(Auto)-induced Equiv y)
| . | | ⊢  Auto(l)↓ = A(l)↓  by Analyze 14
| . | | |
| . | | 14. l1 : Alph*
| . | | 15. l2 : Alph*
| . | | 16. l1 LangOf(Auto)-induced Equiv l2
| . | | ⊢  Auto(l1)↓ = A(l2)↓
| . | | | by Unfold ‘lang_rel‘ 16 THEN Reduce 16 THEN Witness16 nil
| . | | |    THEN
| . | | |    Reduce 16
| . | | |
| . | | 16. LangOf(Auto)(l1) ⟺ LangOf(Auto)(l2)
| . | | ⊢   by Unfold ‘auto_lang‘ 16 THEN Reduce 16
| . | | |
| . | | 16. Auto(l1)↓ ⟺ Auto(l2)↓
| . | | ⊢
| . | | | by FwdThru
| . | | |    Thm* ∀a,b:𝔹. (a ⟺ b) ⇒ a = b    {iff_imp_equal_bool}
| . | | |    [16]
| . | | |
| . | | 17. Auto(l1)↓ = Auto(l2)↓
| . | | ⊢   by RWH (HypC 17) 0
| . | | |
| . | | ⊢  Auto(l2)↓ = A(l2)↓
| . | | | by Unfolds [‘lang_eq‘;‘auto_lang‘] 10 THEN Reduce 10
| . | | |
| . | | 10. ∀l:Alph*. Auto(l)↓ ⟺ A(l)↓
| . | | 11. EquivRel x,y:Alph*. x LangOf(Auto)-induced Equiv y
| . | | 12. EquivRel x,y:Alph*. x LangOf(A)-induced Equiv y
| . | | 13. A ≡ A(λl.A(l)↓)
| . | | 14. l1 : Alph*
| . | | 15. l2 : Alph*
| . | | 16. Auto(l1)↓ ⟺ Auto(l2)↓
```

```
| . | | 17. Auto(l1)↓ = Auto(l2)↓
| . | | ⊢
| . | | | by BackThru
| . | | |    Thm* ∀a,b:𝔹. (a ⟺ b) ⇒ a = b    {iff_imp_equal_bool}
| . | | |
| . | | ⊢  Auto(l2)↓ ⟺ A(l2)↓  by Witness10 l2
| . | |
| . | .\
| . | . .....eq aux .....
| . | . ⊢  x,y:Alph*//(x LangOf(Auto)-induced Equiv y)
| . | .    =
| . | .    x,y:Alph*//(x LangOf(Auto)-induced Equiv y)  by Auto
| . |
| . .\
| . . 14. MinAuto(Auto) = MinAuto(A)
| . . ⊢   by Unfold `min_auto` 14
| . . |
| . . 14. A(λl.Auto(l)↓) = A(λl.A(l)↓)
| . . ⊢   by RWH (HypC 14) 0
| . . |
| . . ⊢  A ≡ A(λl.A(l)↓)
| . . | by Assert
| . . |    (x,y:Alph*//(x LangOf(Auto)-induced Equiv y)
| . . |     =
| . . |     x,y:Alph*//(x LangOf(A)-induced Equiv y))
| . . |
| . . |\
| . . | .....assertion .....
| . . | ⊢  x,y:Alph*//(x LangOf(Auto)-induced Equiv y)
| . . |    =
| . . |    x,y:Alph*//(x LangOf(A)-induced Equiv y)
| . . |   by BackThru
| . . |      Thm* ∀A1:Automata(Alph;S1), A2:Automata(Alph;S2).
| . . |           LangOf(A1) = LangOf(A2) ⇒
| . . |           x,y:Alph*//(x LangOf(A1)-induced Equiv y)
| . . |           =
| . . |           x,y:Alph*//(x LangOf(A2)-induced Equiv y)
| . . |      {lang_eq_imp_quo_eq}
| . . |
| . . .\
| . . . 15. x,y:Alph*//(x LangOf(Auto)-induced Equiv y)
| . . .     =
| . . .     x,y:Alph*//(x LangOf(A)-induced Equiv y)
| . . . ⊢   by RWH (HypC 15) 0
|
.\
. 1. Alph : Type
. 2. St : Type
. 3. Auto : Automata(Alph;St)
. 4. S : Type
. 5. A : Automata(Alph;S)
. 6. Fin(Alph)
. 7. Fin(S)
. 8. Con(A)
. ⊢  EquivRel x,y:Alph*. x LangOf(Auto)-induced Equiv y
```

```
.   by BackThru
.      Thm* ∀L:LangOver(A). EquivRel x,y:A*. x L-induced Equiv y
.      {lang_rel_equi}
END
```