```
lang_auto_is_min  THM

⊢  ∀Auto:Automata(Alph;St)
|  , g:((x,y:Alph*//(x LangOf(Auto)-induced Equiv y))→𝔹).
|  Auto ⊆ A(g)   by Analyze 0 THEN Analyze 0 THEN Analyze 0
|
1. Alph : Type
2. St : Type
3. Auto : Automata(Alph;St)
⊢  ∀g:((x,y:Alph*//(x LangOf(Auto)-induced Equiv y))→𝔹). Auto ⊆ A(g)
| 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
5. g : (x,y:Alph*//(x LangOf(Auto)-induced Equiv y))→𝔹
⊢  Auto ⊆ A(g)   by Unfold `refine` 0
|
6. x : Alph*
7. y : Alph*
8. (Result(Auto)x) = (Result(Auto)y)
⊢  (Result(A(g))x) = (Result(A(g))y)
| by RWH
|    (LemmaC
|     Thm* ∀L:LangOver(Alph), g:((x,y:Alph*//(x L-induced Equiv y))→
|                                𝔹)
|          , l:Alph*.
|          (Result(A(g))l) = l ∈ x,y:Alph*//(x L-induced Equiv y)
|     {lang_auto_compute})
|    0
|
⊢  x = y  by Analyze
|
⊢  x LangOf(Auto)-induced Equiv y
| by Unfold `lang_rel` 0 THEN Reduce 0
|
⊢  ∀z:Alph*. LangOf(Auto)(z @ x) ⟺ LangOf(Auto)(z @ y)  by Analyze 0
|
9. z : Alph*
⊢  LangOf(Auto)(z @ x) ⟺ LangOf(Auto)(z @ y)
| by Unfold `auto_lang` 0 THEN Reduce 0
|
⊢  Auto(z @ x)↓ ⟺ Auto(z @ y)↓
| by Inst
|    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}
|    [Alph;St;Auto;x;y;z]
|
10. (Result(Auto)z @ x) = (Result(Auto)z @ y)
⊢   by RWH (UnfoldC `accept_list`) 0
|
⊢  FinalState(Auto)(Result(Auto)z @ x)
```

```
   ⟺
   FinalState(Auto)(Result(Auto)z @ y)  by RWH (HypC 10) 0
END
```