```
mn_23_lem  THM

⊢  ∀R:(Alph*→Alph*→Prop).
|  Fin(Alph) ⇒
|  (EquivRel x,y:Alph*. x R y) ⇒
|  Fin(x,y:Alph*//(x R y)) ⇒
|  (∀x,y,z:Alph*. (x R y) ⇒ ((z @ x) R (z @ y))) ⇒
|  (∀g:((x,y:Alph*//(x R y))→𝔹), x,y:x,y:Alph*//(x R y). Dec(x Rg y))
| by RepeatFor 7 (Id THEN (Analyze 0))
|
1. Alph : Type
2. R : Alph*→Alph*→Prop
3. Fin(Alph)
4. EquivRel x,y:Alph*. x R y
5. Fin(x,y:Alph*//(x R y))
6. ∀x,y,z:Alph*. (x R y) ⇒ ((z @ x) R (z @ y))
7. g : (x,y:Alph*//(x R y))→𝔹
⊢  ∀x,y:x,y:Alph*//(x R y). Dec(x Rg y)
| by Assert Fin((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y)))
|
|\
| .....assertion .....
| ⊢  Fin((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y)))
| | by Inst Thm* ∀t:T. Fin(T) ⇒ Fin(T×T)    {prod_fin_is_fin}
| |    [x,y:Alph*//(x R y);nil]
| |
| .....wf .....
| ⊢  nil ∈ x,y:Alph*//(x R y)  by Analyze
|
.\
. 8. Fin((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y)))
. ⊢
. | by Assert
. |    (∀a:Alph, x:x,y:Alph*//(x R y). a.x ∈ x,y:Alph*//(x R y))
. |
. |\
. | .....assertion .....
. | ⊢  ∀a:Alph, x:x,y:Alph*//(x R y). a.x ∈ x,y:Alph*//(x R y)
. | | by RepD THEN (Analyze -1) THEN Analyze
. | |
. | 9. a : Alph
. | 10. x1 : Alph*
. | 11. x2 : Alph*
. | 12. x1 R x2
. | ⊢  (a.x1) R (a.x2)  by (InstHyp [x1;x2;[a]] 6) THEN (Reduce -1)
. |
. .\
. . 9. ∀a:Alph, x:x,y:Alph*//(x R y). a.x ∈ x,y:Alph*//(x R y)
. . ⊢
. . | by Inst
. . |    Thm* ∀f:(T→𝔹).
. . |         Fin(T) ⇒ (∃fL:T*. ∀t:T. f(t) ⟺ mem_f(T;t;fL))
. . |    {bool_listify}
. . |    [(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . |    ;λx.x/x1,x2.¬₂(g(x1)) =_b (g(x2))]
```

```
. . |
. . 10. ∃fL:((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y)))*.
. . |   ∀t:((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))).
. . |   (λx.x/x1,x2.¬_2(g(x1)) =_b (g(x2)))(t)
. . |   ⟺
. . |   mem_f((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y));t;fL)
. . ⊢
. . | by Assert
. . |    (<(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . |     ,λa,xy. xy/x,y.<a.x,a.y>>
. . |     ∈ ActionSet(Alph))
. . |    THENA
. . |    Unfold `action_set` 0
. . |
. . |\
. . | 11. a : Alph
. . | 12. xy : (x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . | 13. x : x,y:Alph*//(x R y)
. . | 14. y : x,y:Alph*//(x R y)
. . | 15. xy = <x,y>
. . | ⊢  a.x ∈ x,y:Alph*//(x R y)  by BackThru 9
. . |
. . |\
. . | 11. a : Alph
. . | 12. xy : (x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . | 13. x : x,y:Alph*//(x R y)
. . | 14. y : x,y:Alph*//(x R y)
. . | 15. xy = <x,y>
. . | ⊢  a.y ∈ x,y:Alph*//(x R y)  by BackThru 9
. . |
. . .\
. . . 11. <(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . |   ,λa,xy. xy/x,y.<a.x,a.y>>
. . . |   ∈ ActionSet(Alph)
. . . ⊢
. . . | by Analyze -2
. . . |    THEN
. . . |    Inst
. . . |    Thm* ∀S:ActionSet(Alph), sL:S.car*.
. . . |         Fin(Alph) ⇒
. . . |         Fin(S.car) ⇒
. . . |         (∃TBL:S.car*.
. . . |          ∀s:S.car.
. . . |          mem_f(S.car;s;TBL)
. . . |          ⟺
. . . |          (∃w:Alph*. mem_f(S.car;(S:w←s);sL)))
. . . |    {total_back_listify}
. . . |    [Alph
. . . |    ;<(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . |     ,λa,xy. xy/x,y.<a.x,a.y>>
. . . |    ;fL]
. . . |
. . . |\
. . . | .....antecedent .....
. . . | 10. fL : ((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y)))*
```

```
. . . | 11. ∀t:((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))).
. . . |     (λx.x/x1,x2.¬₂(g(x1)) =_b (g(x2)))(t)
. . . |     ⟺
. . . |     mem_f((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y));t;fL)
. . . | 12. <(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . |     ,λa,xy. xy/x,y.<a.x,a.y>>
. . . |     ∈ ActionSet(Alph)
. . . | ⊢  Fin(<(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . |        ,λa,xy. xy/x,y.<a.x,a.y>>.car)
. . . |   by Unfold `aset_car` 0 THEN Reduce 0 THEN NthHyp 8
. . . |
. . . .\
. . . . 10. fL : ((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y)))*
. . . . 11. ∀t:((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))).
. . . . |   (λx.x/x1,x2.¬₂(g(x1)) =_b (g(x2)))(t)
. . . . |   ⟺
. . . . |   mem_f((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y));t;fL)
. . . . 12. <(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . . |   ,λa,xy. xy/x,y.<a.x,a.y>>
. . . . |   ∈ ActionSet(Alph)
. . . . 13. ∃TBL:<(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . . |        ,λa,xy. xy/x,y.<a.x,a.y>>.car*.
. . . . |
. . . . |   ∀s:<(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . . |      ,λa,xy. xy/x,y.<a.x,a.y>>.car.
. . . . |
. . . . |   mem_f(<(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . . |         ,λa,xy. xy/x,y.<a.x,a.y>>.car;s;TBL)
. . . . |   ⟺
. . . . |   (∃w:Alph*.
. . . . |    mem_f(<(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . . |          ,λa,xy. xy/x,y.
. . . . |           <a.x,a.y>>.car;(<(x,y:Alph*//(x R y))×
. . . . |                             (x,y:Alph*//(x R y))
. . . . |                            ,λa,xy. xy/x,y.
. . . . |                             <a.x,a.y>>:w←s);fL))
. . . . ⊢
. . . . | by Unfold `aset_car` -1 THEN Reduce -1 THEN Reduce -2
. . . . |    THEN
. . . . |    GenExRepD
. . . . |
. . . . 13. TBL : ((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y)))*
. . . . 14. ∀s:((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))).
. . . . |   mem_f((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y));s;TBL)
. . . . |   ⟺
. . . . |   (∃w:Alph*.
. . . . |    mem_f((x,y:Alph*//(x R y))×
. . . . |    (x,y:Alph*//(x R y));(<(x,y:Alph*//(x R y))×
. . . . |                           (x,y:Alph*//(x R y))
. . . . |                          ,λa,xy. xy/x,y.<a.x,a.y>>:w←s);fL))
. . . . 15. x : x,y:Alph*//(x R y)
. . . . 16. y : x,y:Alph*//(x R y)
. . . . ⊢  Dec(x Rg y)
. . . . | by Assert
. . . . |    (∀w:Alph*, s:((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))).
```

```
. . . . |     (<(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . . |      ,λa,xy. xy/x,y.<a.x,a.y>>:w←s)
. . . . |     =
. . . . |     (s/x,y.<w@qx,w@qy>))
. . . . |
. . . . |\
. . . . | .....assertion .....
. . . . | ⊢  ∀w:Alph*, s:((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))).
. . . . | |  (<(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . . | |   ,λa,xy. xy/x,y.<a.x,a.y>>:w←s)
. . . . | |  =
. . . . | |  (s/x,y.<w@qx,w@qy>)  by RepD THEN (ListInd -2)
. . . . | |
. . . . | |\
. . . . | | 17. Alph*
. . . . | | 18. s : (x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . . | | ⊢  (<(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . . | |     ,λa,xy. xy/x,y.<a.x,a.y>>:nil←s)
. . . . | |    =
. . . . | |    (s/x,y.<nil@qx,nil@qy>)
. . . . | |   by RecUnfold ‘maction‘ 0 THEN Analyze -1
. . . . | |      THEN
. . . . | |      Unfold ‘mn_quo_append‘ 0
. . . . | |      THEN
. . . . | |      Reduce 0
. . . . | |
. . . . | .\
. . . . | . 17. Alph*
. . . . | . 18. s : (x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . . | . 19. u : Alph
. . . . | . 20. v : Alph*
. . . . | . 21. (<(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . . | . |     ,λa,xy. xy/x,y.<a.x,a.y>>:v←s)
. . . . | . |    =
. . . . | . |    (s/x,y.<v@qx,v@qy>)
. . . . | . ⊢  (<(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . . | . |    ,λa,xy. xy/x,y.<a.x,a.y>>:u.v←s)
. . . . | . |  =
. . . . | . |  (s/x,y.<u.v@qx,u.v@qy>)
. . . . | . | by RecUnfold ‘maction‘ 0 THEN Analyze 18 THEN Reduce 0
. . . . | . |    THEN
. . . . | . |    Reduce -1
. . . . | . |
. . . . | . 18. s1 : x,y:Alph*//(x R y)
. . . . | . 19. s2 : x,y:Alph*//(x R y)
. . . . | . 20. u : Alph
. . . . | . 21. v : Alph*
. . . . | . 22. (<(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . . | .       ,λa,xy. xy/x,y.<a.x,a.y>>:v←<s1,s2>)
. . . . | .      =
. . . . | .      <v@qs1,v@qs2>
. . . . | . ⊢  ((<(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . . | .       ,λa,xy. xy/x,y.<a.x,a.y>>:v←<s1,s2>)
. . . . | .      /x,y.<u.x,u.y>)
. . . . | .    =
```

```
. . . . | .    <u.v@qs1,u.v@qs2>
. . . . | .   by ((RWH (HypC -1) 0) THEN (Unfold `mn_quo_append` 0)
. . . . | .       THEN
. . . . | .       (Reduce 0)
. . . . | .       THEN
. . . . | .       (Fold `mn_quo_append` 0))
. . . . | .      THEN
. . . . | .      BackThru 9
. . . . |
. . . . .\
. . . . . 17. ∀w:Alph*, s:((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))).
. . . . . |   (<(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . . . |    ,λa,xy. xy/x,y.<a.x,a.y>>:w←s)
. . . . . |   =
. . . . . |   (s/x,y.<w@qx,w@qy>)
. . . . . ⊢
. . . . . | by (RWH (HypC -1) 14) THEN (Thin -1)
. . . . . |    THEN
. . . . . |    (InstHyp [<x,y>] 14)
. . . . . |    THEN
. . . . . |    (Thin 14)
. . . . . |    THEN
. . . . . |    (Reduce -1)
. . . . . |
. . . . . 14. x : x,y:Alph*//(x R y)
. . . . . 15. y : x,y:Alph*//(x R y)
. . . . . 16. mem_f((x,y:Alph*//(x R y))×
. . . . . |   (x,y:Alph*//(x R y));<x,y>;TBL)
. . . . . |   ⟺
. . . . . |   (∃w:Alph*.
. . . . . |    mem_f((x,y:Alph*//(x R y))×
. . . . . |    (x,y:Alph*//(x R y));<w@qx,w@qy>;fL))
. . . . . ⊢
. . . . . | by ((RWH (RevHypC 11) -1) THEN (Thin 11) THEN (Reduce -1))
. . . . . |    THENA
. . . . . |    Reduce 0
. . . . . |
. . . . . 11. <(x,y:Alph*//(x R y))×(x,y:Alph*//(x R y))
. . . . . |   ,λa,xy. xy/x,y.<a.x,a.y>>
. . . . . |   ∈ ActionSet(Alph)
. . . . . 12. TBL : ((x,y:Alph*//(x R y))×(x,y:Alph*//(x R y)))*
. . . . . 13. x : x,y:Alph*//(x R y)
. . . . . 14. y : x,y:Alph*//(x R y)
. . . . . 15. mem_f((x,y:Alph*//(x R y))×
. . . . . |   (x,y:Alph*//(x R y));<x,y>;TBL)
. . . . . |   ⟺
. . . . . |   (∃w:Alph*. ¬2(g(w@qx)) =b (g(w@qy)))
. . . . . ⊢
. . . . . | by ((Unfold `lquo_rel` 0 THEN Reduce 0
. . . . . |      THEN
. . . . . |      Inst
. . . . . |      Thm* ∀P:(T→Prop).
. . . . . |           (∀x:T. Dec(P(x))) & Dec(∃x:T. ¬P(x)) ⇒
. . . . . |           Dec(∀x:T. P(x))
. . . . . |      {auto2_lemma_0}
```

```
. . . . . . |      [Alph*;λz.g(z@_q x) ⟺ g(z@_q y)])
. . . . . . |     THEN
. . . . . . |     (Reduce -1))
. . . . . . |    THEN
. . . . . . |    Try (Complete Auto)
. . . . . . |
. . . . . . .....antecedent .....
. . . . . . ⊢  (∀x@0:Alph*. Dec((λz.g(z@_q x) ⟺ g(z@_q y))(x@0)))
. . . . . . |  & Dec(∃x@0:Alph*. ¬(λz.g(z@_q x) ⟺ g(z@_q y))(x@0))
. . . . . . | by Reduce 0 THEN Analyze 0
. . . . . . |
. . . . . . |\
. . . . . . | ⊢  ∀x@0:Alph*. Dec(g(x@0@_q x) ⟺ g(x@0@_q y))  by Auto
. . . . . . |
. . . . . . .\
. . . . . . . ⊢  Dec(∃x@0:Alph*. ¬(g(x@0@_q x) ⟺ g(x@0@_q y)))
. . . . . . . | by (Assert
. . . . . . . |     (∀w:Alph*.
. . . . . . . |      ¬(g(w@_q x) ⟺ g(w@_q y)) ⟺ ¬_2(g(w@_q x)) =_b (g(w@_q y))))
. . . . . . . |    THEN
. . . . . . . |    (RWH (HypC -1) 0)
. . . . . . . |    THEN
. . . . . . . |    (Thin -1)
. . . . . . . |
. . . . . . . |\
. . . . . . . | 15. mem_f((x,y:Alph*//(x R y))×
. . . . . . . | |    (x,y:Alph*//(x R y));<x,y>;TBL)
. . . . . . . | |    ⇒
. . . . . . . | |    (∃w:Alph*. ¬_2(g(w@_q x)) =_b (g(w@_q y)))
. . . . . . . | 16. mem_f((x,y:Alph*//(x R y))×
. . . . . . . | |    (x,y:Alph*//(x R y));<x,y>;TBL) ⇐ (∃w:Alph*.
. . . . . . . | |    ¬_2(g(w@_q x)) =_b (g(w@_q y)))
. . . . . . . | 17. w : Alph*
. . . . . . . | 18. ¬(g(w@_q x) ⟺ g(w@_q y))
. . . . . . . | ⊢  ¬_2(g(w@_q x)) =_b (g(w@_q y))
. . . . . . . | | by (RWH assert_pushdownC 0) THEN (Analyze 0)
. . . . . . . | |    THEN
. . . . . . . | |    (Analyze -2)
. . . . . . . | |
. . . . . . . | 18. (g(w@_q x)) =_b (g(w@_q y))
. . . . . . . | ⊢  g(w@_q x) ⟺ g(w@_q y)
. . . . . . . |   by (FwdThru Thm* ∀b:𝔹. b ⇒ b = true_2    {assert_elim}
. . . . . . . |       [-1])
. . . . . . . |      THEN
. . . . . . . |      (FwdThru
. . . . . . . |       Thm* ∀p,q:𝔹. p =_b q = true_2 ⟺ p = q    {beq_eq}
. . . . . . . |       [-1])
. . . . . . . |      THEN
. . . . . . . |      (RWH (HypC -1) 0)
. . . . . . . |
. . . . . . . |\
. . . . . . . | 15. mem_f((x,y:Alph*//(x R y))×
. . . . . . . | |    (x,y:Alph*//(x R y));<x,y>;TBL)
. . . . . . . | |    ⇒
. . . . . . . | |    (∃w:Alph*. ¬_2(g(w@_q x)) =_b (g(w@_q y)))
```

```
. . . . . . . | 16. mem_f((x,y:Alph*//(x R y))×
. . . . . . . | |    (x,y:Alph*//(x R y));<x,y>;TBL) ⇐ (∃w:Alph*.
. . . . . . . | |    ¬2(g(w@qx)) =b (g(w@qy)))
. . . . . . . | 17. w : Alph*
. . . . . . . | 18. ¬2(g(w@qx)) =b (g(w@qy))
. . . . . . . | ⊢  ¬(g(w@qx) ⟺ g(w@qy))
. . . . . . . | | by (RWH assert_pushdownC -1) THEN (Analyze 0)
. . . . . . . | |    THEN
. . . . . . . | |    (Analyze -2)
. . . . . . . | |
. . . . . . . | 18. g(w@qx) ⟺ g(w@qy)
. . . . . . . | ⊢  (g(w@qx)) =b (g(w@qy))
. . . . . . . |   by (FwdThru
. . . . . . . |        Thm* ∀a,b:𝔹. (a ⟺ b) ⇒ a = b
. . . . . . . |        {iff_imp_equal_bool}
. . . . . . . |        [-1])
. . . . . . . |      THEN
. . . . . . . |      (RWH (HypC -1) 0)
. . . . . . . |      THEN
. . . . . . . |      (RWH (LemmaC Thm* ∀p:𝔹. p =b p = true2    {beq_id})
. . . . . . . |       0)
. . . . . . . |      THEN
. . . . . . . |      (BackThru Thm* true2    {assert_of_tt})
. . . . . . . |
. . . . . . . .\
. . . . . . . . ⊢  Dec(∃x@0:Alph*. ¬2(g(x@0@qx)) =b (g(x@0@qy)))
. . . . . . . .   by (RWH (RevHypC -1) 0)
. . . . . . . .      THEN
. . . . . . . .      (BackThru
. . . . . . . .       Thm* ∀s:S, l:S*. Fin(S) ⇒ Dec(mem_f(S;s;l))
. . . . . . . .       {mem_f_dec})
END
```