```
reach_aux  THM

⊢  ∀S:ActionSet(Alph), si:S.car.
|  Fin(S.car) ⇒
|  Fin(Alph) ⇒
|  (∃RL:S.car*.
|   ∀s:S.car. (∃w:Alph*. (S:w←si) = s) ⟺ mem_f(S.car;s;RL))  by RepD
|
1. Alph : Type
2. S : ActionSet(Alph)
3. si : S.car
4. Fin(S.car)
5. Fin(Alph)
⊢  ∃RL:S.car*. ∀s:S.car. (∃w:Alph*. (S:w←si) = s) ⟺ mem_f(S.car;s;RL)
| by (Analyze -1
|     THEN
|     RWH
|     (LemmaC
|      Thm* (∃f:(A→B). Bij(A; B; f)) ⟺ (A ∼ B)
|      {bij_iff_1_1_corr})
|     -1)
|    THEN
|    (Analyze -1)
|    THEN
|    (Analyze -1)
|
5. n : ℕ
6. f : ℕn→Alph
7. g : Alph→ℕn
8. InvFuns(ℕn; Alph; f; g)
⊢   by Assert Fin(S.car) THENA NthHyp 4
|
9. Fin(S.car)
⊢
| by (Analyze -1
|     THEN
|     RWH
|     (LemmaC
|      Thm* (∃f:(A→B). Bij(A; B; f)) ⟺ (A ∼ B)
|      {bij_iff_1_1_corr})
|     -1)
|    THEN
|    (Analyze -1)
|    THEN
|    (Analyze -1)
|
9. n1 : ℕ
10. f1 : ℕn1→S.car
11. g1 : S.car→ℕn1
12. InvFuns(ℕn1; S.car; f1; g1)
⊢
| by (Inst
|     Thm* ∀S:ActionSet(Alph), si:S.car, nn:ℕ, f:(ℕnn→Alph)
|          , g:(Alph→ℕnn).
|          Fin(S.car) ⇒
```

```
|           InvFuns(ℕnn; Alph; f; g) ⇒
|           (∀n:ℕ.
|            ∃RL:{y:{x:(S.car*)| 0<||x|| & ||x||≤n+1 }
|                 | y[(||y||-1)] = si }.
|
|            (∀s:S.car. (∃w:Alph*. (S:w←si) = s) ⟺ mem_f(S.car;s;RL))
|            ∨ ||RL|| = n+1
|              & (∀i:ℕ||RL||, j:ℕi. ¬RL[i] = RL[j])
|              & (∀s:S.car.
|                  mem_f(S.car;s;RL) ⇒ (∃w:Alph*. (S:w←si) = s))
|              & (∀k:ℕ.
|                  k≤nn ⇒
|                  (∃RLa:S.car*.
|                   (∀i:{1..||RL||⁻}, a:Alph.
|                    mem_f(S.car;S.act(a,RL[i]);RL)
|                    ∨ mem_f(S.car;S.act(a,RL[i]);RLa))
|                   & (∀a:Alph.
|                      g(a)<k ⇒
|                      mem_f(S.car;S.act(a,hd(RL));RL)
|                      ∨ mem_f(S.car;S.act(a,hd(RL));RLa))
|                   & (∀s:S.car.
|                      mem_f(S.car;s;RLa) ⇒
|                      (∃w:Alph*. (S:w←si) = s)))))
|     {reach_lemma}
|     [Alph;S;si;n;f;g;n1])
|    THEN
|    GenExRepD
|
|\
| 13. RL : {y:{x:(S.car*)| 0<||x|| & ||x||≤n1+1 }| y[(||y||-1)] = si }
| 14. ∀s:S.car. (∃w:Alph*. (S:w←si) = s) ⟺ mem_f(S.car;s;RL)
| ⊢   by InstConcl [RL]
|
.\
. 13. RL : {y:{x:(S.car*)| 0<||x|| & ||x||≤n1+1 }| y[(||y||-1)] = si }
. 14. ||RL|| = n1+1
. 15. ∀i:ℕ||RL||, j:ℕi. ¬RL[i] = RL[j]
. 16. ∀s:S.car. mem_f(S.car;s;RL) ⇒ (∃w:Alph*. (S:w←si) = s)
. 17. ∀k:ℕ.
. |   k≤n ⇒
. |   (∃RLa:S.car*.
. |    (∀i:{1..||RL||⁻}, a:Alph.
. |     mem_f(S.car;S.act(a,RL[i]);RL)
. |     ∨ mem_f(S.car;S.act(a,RL[i]);RLa))
. |    & (∀a:Alph.
. |       g(a)<k ⇒
. |       mem_f(S.car;S.act(a,hd(RL));RL)
. |       ∨ mem_f(S.car;S.act(a,hd(RL));RLa))
. |    & (∀s:S.car. mem_f(S.car;s;RLa) ⇒ (∃w:Alph*. (S:w←si) = s)))
. ⊢
. | by (Thin -1)
. |    THEN
. |    (Inst
. |     Thm* ∀n:{1...}, f:(ℕ(n+1)→ℕn). ∃i:ℕ(n+1), j:ℕi. f(i) = f(j)
. |     {phole_aux}
```

```
. |     [n1;λx.g1(RL[x])])
. |    THEN
. |    ExRepD
. |
. |\
. | .....set predicate .....
. | 16. ∀s:S.car. mem_f(S.car;s;RL) ⇒ (∃w:Alph*. (S:w←si) = s)
. | ⊢  1≤n1  by Assert (g1(si) ∈ ℕn1)
. |
. .\
. . 17. i : ℕ(n1+1)
. . 18. j : ℕi
. . 19. (λx.g1(RL[x]))(i) = (λx.g1(RL[x]))(j)
. . ⊢   by Reduce -1 THEN InstHyp [i;j] 15
. . |
. . 19. g1(RL[i]) = g1(RL[j])
. . 20. ¬RL[i] = RL[j]
. . ⊢   by Analyze -1
. . |
. . 19. g1(RL[i]) = g1(RL[j])
. . ⊢  RL[i] = RL[j]  by Assert (f1(g1(RL[i])) = f1(g1(RL[j])))
. . |
. . |\
. . | .....assertion .....
. . | ⊢  f1(g1(RL[i])) = f1(g1(RL[j]))  by RWH (HypC -1) 0
. . |
. . .\
. . . 20. f1(g1(RL[i])) = f1(g1(RL[j]))
. . . ⊢   by RWH add_composeC -1 THEN Analyze 12
. . . |
. . . 12. g1 o f1 = Id
. . . 13. f1 o g1 = Id
. . . 14. RL :
. . .     {y:{x:(S.car*)| 0<||x|| & ||x||≤n1+1 }| y[(||y||-1)] = si }
. . . 15. ||RL|| = n1+1
. . . 16. ∀i:ℕ||RL||, j:ℕi. ¬RL[i] = RL[j]
. . . 17. ∀s:S.car. mem_f(S.car;s;RL) ⇒ (∃w:Alph*. (S:w←si) = s)
. . . 18. i : ℕ(n1+1)
. . . 19. j : ℕi
. . . 20. g1(RL[i]) = g1(RL[j])
. . . 21. (f1 o g1)(RL[i]) = (f1 o g1)(RL[j])
. . . ⊢   by (RWH (HypC 13) -1) THEN (Reduce -1)
END
```