reach_lemma 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)))))
| by RepD
|
1. Alph : Type
2. S : ActionSet(Alph)
3. si : S.car
4. nn : ℕ
5. f : ℕnn→Alph
6. g : Alph→ℕnn
7. Fin(S.car)
8. InvFuns(ℕnn; Alph; f; g)
9. 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))))
| by NatInd 9
|
|\
| .....basecase .....
```

```
| 8. InvFuns(ℕnn; Alph; f; g)
| ⊢  ∃RL:{y:{x:(S.car*)| 0<||x|| & ||x||≤0+1 }| y[(||y||-1)] = si }.
| |  (∀s:S.car. (∃w:Alph*. (S:w←si) = s) ⟺ mem_f(S.car;s;RL))
| |  ∨ ||RL|| = 0+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))))
| | by (InstConcl [[si]]) THEN (Sel 2 (Analyze 0))
| |
| |\
| | .....wf .....
| | ⊢  [si]
| |    ∈ {y:{x:(S.car*)| 0<||x|| & ||x||≤0+1 }| y[(||y||-1)] = si }
| |   by Analyze THENL [Analyze;Reduce 0;Analyze -1] THEN Reduce 0
| |
| |\
| | ⊢  ||[si]|| = 0+1
| | |  & (∀i:ℕ||[si]||, j:ℕi. ¬[si][i] = [si][j])
| | |  & (∀s:S.car. mem_f(S.car;s;[si]) ⇒ (∃w:Alph*. (S:w←si) = s))
| | |  & (∀k:ℕ.
| | |     k≤nn ⇒
| | |     (∃RLa:S.car*.
| | |      (∀i:{1..||[si]||⁻}, a:Alph.
| | |       mem_f(S.car;S.act(a,[si][i]);[si])
| | |       ∨ mem_f(S.car;S.act(a,[si][i]);RLa))
| | |      & (∀a:Alph.
| | |         g(a)<k ⇒
| | |         mem_f(S.car;S.act(a,hd([si]));[si])
| | |         ∨ mem_f(S.car;S.act(a,hd([si]));RLa))
| | |      & (∀s:S.car.
| | |         mem_f(S.car;s;RLa) ⇒ (∃w:Alph*. (S:w←si) = s))))
| | | by Reduce 0
| | |
| | |\
| | | 9. s : S.car
| | | 10. si = s ∨ False

| | | ⊢  ∃w:Alph*. (S:w←si) = s
| | |   by (InstConcl [nil]) THEN (RecUnfold ‘maction‘ 0)
| | |      THEN
| | |      (Reduce 0)
| | |      THEN
| | |      (Analyze 10)
| | |
| | .\
```

```
| | . 9. k : ℕ
| | . 10. k≤nn
| | . ⊢  ∃RLa:S.car*.
| | . |  (∀i:{1..1⁻}, a:Alph.
| | . |   si = S.act(a,[si][i]) ∨ False
| | . |   ∨ mem_f(S.car;S.act(a,[si][i]);RLa))
| | . |  & (∀a:Alph.
| | . |     g(a)<k ⇒
| | . |     si = S.act(a,si) ∨ False ∨ mem_f(S.car;S.act(a,si);RLa))
| | . |  & (∀s:S.car. mem_f(S.car;s;RLa) ⇒ (∃w:Alph*. (S:w←si) = s))
| | . | by NatInd 9
| | . |
| | . |\
| | . | .....basecase .....
| | . | 8. InvFuns(ℕnn; Alph; f; g)
| | . | ⊢  0≤nn ⇒
| | . | |  (∃RLa:S.car*.
| | . | |   (∀i:{1..1⁻}, a:Alph.
| | . | |    si = S.act(a,[si][i]) ∨ False
| | . | |    ∨ mem_f(S.car;S.act(a,[si][i]);RLa))
| | . | |   & (∀a:Alph.
| | . | |      g(a)<0 ⇒
| | . | |      si = S.act(a,si) ∨ False
| | . | |      ∨ mem_f(S.car;S.act(a,si);RLa))
| | . | |   & (∀s:S.car.
| | . | |      mem_f(S.car;s;RLa) ⇒ (∃w:Alph*. (S:w←si) = s)))
| | . | | by (Analyze 0) THEN (InstConcl [nil])
| | . | |
| | . | 9. 0≤nn
| | . | 10. s : S.car
| | . | 11. mem_f(S.car;s;nil)
| | . | ⊢  ∃w:Alph*. (S:w←si) = s  by Reduce -1
| | . |
| | . .\
| | . . .....upcase .....
| | . . 9. k : ℤ
| | . . 10. 0<k
| | . . 11. k-1≤nn ⇒
| | . . |   (∃RLa:S.car*.
| | . . |    (∀i:{1..1⁻}, a:Alph.
| | . . |     si = S.act(a,[si][i]) ∨ False
| | . . |     ∨ mem_f(S.car;S.act(a,[si][i]);RLa))
| | . . |    & (∀a:Alph.
| | . . |       g(a)<k-1 ⇒
| | . . |       si = S.act(a,si) ∨ False
| | . . |       ∨ mem_f(S.car;S.act(a,si);RLa))
| | . . |    & (∀s:S.car.
| | . . |       mem_f(S.car;s;RLa) ⇒ (∃w:Alph*. (S:w←si) = s)))
| | . . ⊢  k≤nn ⇒
| | . . |  (∃RLa:S.car*.
| | . . |   (∀i:{1..1⁻}, a:Alph.
| | . . |    si = S.act(a,[si][i]) ∨ False
| | . . |    ∨ mem_f(S.car;S.act(a,[si][i]);RLa))
| | . . |   & (∀a:Alph.
| | . . |      g(a)<k ⇒
```

```
| | . . |      si = S.act(a,si) ∨ False
| | . . |      ∨ mem_f(S.car;S.act(a,si);RLa))
| | . . |   & (∀s:S.car.
| | . . |      mem_f(S.car;s;RLa) ⇒ (∃w:Alph*. (S:w←si) = s)))
| | . . | by (Analyze 0) THEN (Analyze 11) THEN ExRepD
| | . . |
| | . . 11. k≤nn
| | . . 12. RLa : S.car*
| | . . 13. ∀i:{1..1⁻}, a:Alph.
| | . . |   si = S.act(a,[si][i]) ∨ False
| | . . |   ∨ mem_f(S.car;S.act(a,[si][i]);RLa)
| | . . 14. ∀a:Alph.
| | . . |   g(a)<k-1 ⇒
| | . . |   si = S.act(a,si) ∨ False ∨ mem_f(S.car;S.act(a,si);RLa)
| | . . 15. ∀s:S.car. mem_f(S.car;s;RLa) ⇒ (∃w:Alph*. (S:w←si) = s)
| | . . ⊢  ∃RLa:S.car*.
| | . . |  (∀i:{1..1⁻}, a:Alph.
| | . . |   si = S.act(a,[si][i]) ∨ False
| | . . |   ∨ mem_f(S.car;S.act(a,[si][i]);RLa))
| | . . |  & (∀a:Alph.
| | . . |     g(a)<k ⇒
| | . . |     si = S.act(a,si) ∨ False ∨ mem_f(S.car;S.act(a,si);RLa))
| | . . |  & (∀s:S.car.
| | . . |     mem_f(S.car;s;RLa) ⇒ (∃w:Alph*. (S:w←si) = s))
| | . . | by InstConcl [S.act(f(k-1),si).RLa]
| | . . |
| | . . |\
| | . . | 16. a : Alph
| | . . | 17. g(a)<k
| | . . | ⊢  si = S.act(a,si) ∨ False
| | . . | |  ∨ mem_f(S.car;S.act(a,si);S.act(f(k-1),si).RLa)
| | . . | | by (Reduce 0) THEN (Decide (g(a)<k-1))
| | . . | |
| | . . | |\
| | . . | | 18. g(a)<k-1
| | . . | | ⊢  si = S.act(a,si) ∨ False
| | . . | |    ∨ S.act(f(k-1),si) = S.act(a,si)
| | . . | |    ∨ mem_f(S.car;S.act(a,si);RLa)
| | . . | |   by (FwdThru 14 [-1]) THEN ProveProp
| | . . | |
| | . . | .\
| | . . | . 18. ¬g(a)<k-1
| | . . | . ⊢  si = S.act(a,si) ∨ False
| | . . | . |  ∨ S.act(f(k-1),si) = S.act(a,si)
| | . . | . |  ∨ mem_f(S.car;S.act(a,si);RLa)  by Assert (f(k-1) = a)
| | . . | . |
| | . . | . |\
| | . . | . | .....assertion .....
| | . . | . | ⊢  f(k-1) = a  by Assert (g(a) = k-1)
| | . . | . | |
| | . . | . | 19. g(a) = k-1
| | . . | . | ⊢
| | . . | . | | by (Analyze 8 THEN RWH (RevHypC -1) 0)
| | . . | . | |    THEN
| | . . | . | |    (RWH add_composeC 0)
```

```
| | . . | . | |
| | . . | . | 8. g o f = Id
| | . . | . | 9. f o g = Id
| | . . | . | 10. k : ℤ
| | . . | . | 11. 0<k
| | . . | . | 12. k≤nn
| | . . | . | 13. RLa : S.car*
| | . . | . | 14. ∀i:{1..1⁻}, a:Alph.
| | . . | . |     si = S.act(a,[si][i]) ∨ False
| | . . | . |     ∨ mem_f(S.car;S.act(a,[si][i]);RLa)
| | . . | . | 15. ∀a:Alph.
| | . . | . |     g(a)<k-1 ⇒
| | . . | . |     si = S.act(a,si) ∨ False
| | . . | . |     ∨ mem_f(S.car;S.act(a,si);RLa)
| | . . | . | 16. ∀s:S.car.
| | . . | . |     mem_f(S.car;s;RLa) ⇒ (∃w:Alph*. (S:w←si) = s)
| | . . | . | 17. a : Alph
| | . . | . | 18. g(a)<k
| | . . | . | 19. ¬g(a)<k-1
| | . . | . | 20. g(a) = k-1
| | . . | . | ⊢  (f o g)(a) = a  by (RWH (HypC 9) 0) THEN (Reduce 0)
| | . . | . |
| | . . | . .\
| | . . | . . 19. f(k-1) = a
| | . . | . . ⊢   by (Sel 2 (Analyze 0)) THEN (RWH (HypC -1) 0)
| | . . |
| | . . .\
| | . . . 16. s : S.car
| | . . . 17. mem_f(S.car;s;S.act(f(k-1),si).RLa)
| | . . . ⊢  ∃w:Alph*. (S:w←si) = s  by Reduce -1 THEN Analyze -1
| | . . . |
| | . . . |\
| | . . . | 17. S.act(f(k-1),si) = s
| | . . . | ⊢
| | . . . |   by (InstConcl [[f(k-1)]])
| | . . . |      THEN
| | . . . |      (RepeatFor 2 (RecUnfold `maction` 0))
| | . . . |      THEN
| | . . . |      (Reduce 0)
| | . . . |
| | . . . . .\
| | . . . . 17. mem_f(S.car;s;RLa)
| | . . . . ⊢   by BackThru 15
| |
| |\
| | 9. RL : {y:{x:(S.car*)| 0<||x|| & ||x||≤0+1 }| y[(||y||-1)] = si }
| | 10. k : ℕ
| | 11. k≤nn
| | 12. S.car*
| | 13. a : Alph
| | 14. g(a)<k
| | ⊢  ||RL||≥1
| |   by Analyze 9 THEN Analyze 9 THEN UnhideSqStableHyp 11
| |
| .\
```

```
| . 9. RL : {y:{x:(S.car*)| 0<||x|| & ||x||≤0+1 }| y[(||y||-1)] = si }
| . 10. k : ℕ
| . 11. k≤nn
| . 12. S.car*
| . 13. a : Alph
| . 14. g(a)<k
| . ⊢  ||RL||≥1
| .   by Analyze 9 THEN Analyze 9 THEN UnhideSqStableHyp 11
|
.\
. .....upcase .....
. 9. n : ℤ
. 10. 0<n
. 11. ∃RL:{y:{x:(S.car*)| 0<||x|| & ||x||≤n-1+1 }
. |         | y[(||y||-1)] = si }.
. |
. |   (∀s:S.car. (∃w:Alph*. (S:w←si) = s) ⟺ mem_f(S.car;s;RL))
. |   ∨ ||RL|| = n-1+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))))
. ⊢   by Analyze 11 THEN Analyze 12
. |
. |\
. | 11. RL :
. | |   {y:{x:(S.car*)| 0<||x|| & ||x||≤n-1+1 }| y[(||y||-1)] = si }
. | 12. ∀s:S.car. (∃w:Alph*. (S:w←si) = s) ⟺ mem_f(S.car;s;RL)
. | ⊢   by InstConcl [RL]
. | |
. | |\
. | | 11. RL : S.car*
. | | 12. 0<||RL||
. | | 13. ||RL||≤n-1+1
. | | 14. RL[(||RL||-1)] = si
. | | 15. ∀s:S.car. (∃w:Alph*. (S:w←si) = s) ⟺ mem_f(S.car;s;RL)
. | | 16. y : {x:(S.car*)| 0<||x|| & ||x||≤n+1 }
. | | ⊢  0≤||y||-1  by Analyze -1
. | |
. | |\
. | | ⊢  (∀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))))
. | |   by Sel 1 (Analyze 0) THENA (Analyze 11 THEN Analyze 11)
. | |
. | |\
. | | 13. R1 :
. | |     {y:{x:(S.car*)| 0<||x|| & ||x||≤n+1 }| y[(||y||-1)] = si }
. | | 14. k : ℕ
. | | 15. k≤nn
. | | 16. S.car*
. | | 17. a : Alph
. | | 18. g(a)<k
. | | ⊢  ||R1||≥1
. | |   by Analyze 13 THEN Analyze 13 THEN UnhideSqStableHyp 15
. | |
. | .\
. | . 13. R1 :
. | .     {y:{x:(S.car*)| 0<||x|| & ||x||≤n+1 }| y[(||y||-1)] = si }
. | . 14. k : ℕ
. | . 15. k≤nn
. | . 16. S.car*
. | . 17. a : Alph
. | . 18. g(a)<k
. | . ⊢  ||R1||≥1
. | .   by Analyze 13 THEN Analyze 13 THEN UnhideSqStableHyp 15
. |
. .\
. . 11. RL :
. . |   {y:{x:(S.car*)| 0<||x|| & ||x||≤n-1+1 }| y[(||y||-1)] = si }
. . 12. ||RL|| = n-1+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))))
. . ⊢   by (ExRepD THEN InstHyp [nn] -1) THEN (Analyze -1)
. . |
```

```
. . 12. ||RL|| = n-1+1
. . 13. ∀i:ℕ||RL||, j:ℕi. ¬RL[i] = RL[j]
. . 14. ∀s:S.car. mem_f(S.car;s;RL) ⇒ (∃w:Alph*. (S:w←si) = s)
. . 15. ∀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)))
. . 16. RLa : S.car*
. . 17. ∀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)
. . 18. ∀a:Alph.
. . |   g(a)<nn ⇒
. . |   mem_f(S.car;S.act(a,hd(RL));RL)
. . |   ∨ mem_f(S.car;S.act(a,hd(RL));RLa)
. . 19. ∀s:S.car. mem_f(S.car;s;RLa) ⇒ (∃w:Alph*. (S:w←si) = s)
. . ⊢
. . | by (Inst
. . |     Thm* ∀L,La:T*.
. . |          Fin(T) ⇒
. . |          (∃La':T*.
. . |           (∀t:T. mem_f(T;t;La) ⇒ mem_f(T;t;L) ∨ mem_f(T;t;La'))
. . |           & (∀t:T. mem_f(T;t;La') ⇒ mem_f(T;t;La))
. . |           & (||La'||≥1 ⇒ ¬mem_f(T;hd(La');L)))
. . |     {shorten_list}
. . |     [S.car;RL;RLa])
. . |    THEN
. . |    ExRepD
. . |
. . 20. La' : S.car*
. . 21. ∀t:S.car.
. . |   mem_f(S.car;t;RLa) ⇒ mem_f(S.car;t;RL) ∨ mem_f(S.car;t;La')
. . 22. ∀t:S.car. mem_f(S.car;t;La') ⇒ mem_f(S.car;t;RLa)
. . 23. ||La'||≥1 ⇒ ¬mem_f(S.car;hd(La');RL)
. . ⊢   by Decide (||La'||≥1)
. . |
. . |\
. . | 24. ||La'||≥1
. . | ⊢   by InstConcl [hd(La').RL]
. . | |
. . | |\
. . | | .....wf .....
. . | | ⊢  hd(La').RL
. . | | |  ∈ {y:{x:(S.car*)| 0<||x|| & ||x||≤n+1 }
. . | | |      | y[(||y||-1)] = si }
. . | | | by Analyze 11 THEN Analyze 11 THEN Analyze
. . | | |    THENL
. . | | |    [Analyze;Reduce 0;Analyze -1]
```

```
. . | | |     THEN
. . | | |     Reduce 0
. . | | |
. . | | 11. RL : S.car*
. . | | 12. 0<||RL||
. . | | 13. ||RL||≤n-1+1
. . | | 14. RL[(||RL||-1)] = si
. . | | 15. ||RL|| = n-1+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. ∀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)))
. . | | 19. RLa : S.car*
. . | | 20. ∀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)
. . | | 21. ∀a:Alph.
. . | | |   g(a)<nn ⇒
. . | | |   mem_f(S.car;S.act(a,hd(RL));RL)
. . | | |   ∨ mem_f(S.car;S.act(a,hd(RL));RLa)
. . | | 22. ∀s:S.car. mem_f(S.car;s;RLa) ⇒ (∃w:Alph*. (S:w←si) = s)
. . | | 23. La' : S.car*
. . | | 24. ∀t:S.car.
. . | | |   mem_f(S.car;t;RLa) ⇒
. . | | |   mem_f(S.car;t;RL) ∨ mem_f(S.car;t;La')
. . | | 25. ∀t:S.car. mem_f(S.car;t;La') ⇒ mem_f(S.car;t;RLa)
. . | | 26. ||La'||≥1 ⇒ ¬mem_f(S.car;hd(La');RL)
. . | | 27. ||La'||≥1
. . | | ⊢  (hd(La').RL)[(||RL||+1-1)] = si
. . | | | by ((ArithSimp 14) THEN (Unfold ‘select‘ 0)
. . | | |      THEN
. . | | |      (RecUnfold ‘nth_tl‘ 0))
. . | | |     THEN
. . | | |     Reduce 0
. . | | |
. . | | 14. RL[(-1+||RL||)] = si
. . | | 15. ||RL|| = n-1+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. ∀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)))
. . | | 19. RLa : S.car*
. . | | 20. ∀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)
. . | | 21. ∀a:Alph.
. . | | |   g(a)<nn ⇒
. . | | |   mem_f(S.car;S.act(a,hd(RL));RL)
. . | | |   ∨ mem_f(S.car;S.act(a,hd(RL));RLa)
. . | | 22. ∀s:S.car. mem_f(S.car;s;RLa) ⇒ (∃w:Alph*. (S:w←si) = s)
. . | | 23. La' : S.car*
. . | | 24. ∀t:S.car.
. . | | |   mem_f(S.car;t;RLa) ⇒
. . | | |   mem_f(S.car;t;RL) ∨ mem_f(S.car;t;La')
. . | | 25. ∀t:S.car. mem_f(S.car;t;La') ⇒ mem_f(S.car;t;RLa)
. . | | 26. ||La'||≥1 ⇒ ¬mem_f(S.car;hd(La');RL)
. . | | 27. ||La'||≥1
. . | | ⊢  hd(if ||RL||+1-1≤z0→ hd(La').RL
. . | | |     else nth_tl(||RL||+1-1-1;RL) fi)
. . | | |  =
. . | | |  si  by SplitOnConclITE
. . | | |
. . | | .....falsecase .....
. . | | 28. 0<||RL||+1-1
. . | | ⊢  hd(nth_tl(||RL||+1-1-1;RL)) = si
. . | |   by Fold `select` 0 THEN ArithSimp 0
. . | |
. . | |\
. . | | ⊢  (∀s:S.car.
. . | | |   (∃w:Alph*. (S:w←si) = s) ⟺ mem_f(S.car;s;hd(La').RL))
. . | | |  ∨ ||hd(La').RL|| = n+1
. . | | |    & (∀i:ℕ||hd(La').RL||, j:ℕi.
. . | | |        ¬(hd(La').RL)[i] = (hd(La').RL)[j])
. . | | |    & (∀s:S.car.
. . | | |        mem_f(S.car;s;hd(La').RL) ⇒ (∃w:Alph*. (S:w←si) = s))
. . | | |    & (∀k:ℕ.
. . | | |        k≤nn ⇒
. . | | |        (∃RLa:S.car*.
. . | | |         (∀i:{1..||hd(La').RL||⁻}, a:Alph.
. . | | |          mem_f(S.car;S.act(a,(hd(La').RL)[i]);hd(La').RL)
. . | | |          ∨ mem_f(S.car;S.act(a,(hd(La').RL)[i]);RLa))
. . | | |         & (∀a:Alph.
. . | | |            g(a)<k ⇒
. . | | |            mem_f(S.car;S.act(a,hd((hd(La').RL)));hd(La').RL)
. . | | |            ∨ mem_f(S.car;S.act(a,hd((hd(La').RL)));RLa))
. . | | |         & (∀s:S.car.
. . | | |            mem_f(S.car;s;RLa) ⇒ (∃w:Alph*. (S:w←si) = s))))
. . | | | by Reduce 0 THEN Sel 2 (Analyze 0)
. . | | |
. . | | |\
```

```
. . | | | 25. i : ℕ(||RL||+1)
. . | | | 26. j : ℕi
. . | | | ⊢  ¬(hd(La').RL)[i] = (hd(La').RL)[j]  by Decide (j = 0)
. . | | | |
. . | | | |\
. . | | | | 27. j = 0
. . | | | | ⊢
. . | | | | | by (RWH (HypC -1) 0) THEN (Unfold `select` 0)
. . | | | | |    THEN
. . | | | | |    (RecUnfold `nth_tl` 0)
. . | | | | |    THEN
. . | | | | |    (Reduce 0)
. . | | | | |
. . | | | | |\
. . | | | | | ⊢  i<||hd(La').RL||  by Reduce 0
. . | | | | |
. . | | | | |\
. . | | | | | ⊢  j<||hd(La').RL||  by Reduce 0
. . | | | | |
. . | | | | .\
. . | | | | . ⊢  ¬hd(if i≤_{2}0→ hd(La').RL else nth_tl(i-1;RL) fi)
. . | | | | . |    =
. . | | | | . |    hd(La')  by SplitOnConclITE
. . | | | | . |
. . | | | | . .....falsecase .....
. . | | | | . 28. 0<i
. . | | | | . ⊢  ¬hd(nth_tl(i-1;RL)) = hd(La')
. . | | | | . | by (Fold `select` 0 THEN Analyze 0)
. . | | | | . |    THEN
. . | | | | . |    (FwdThru 23 [24])
. . | | | | . |
. . | | | | . 29. RL[(i-1)] = hd(La')
. . | | | | . 30. ¬mem_f(S.car;hd(La');RL)
. . | | | | . ⊢  False  by (RWH (RevHypC -2) -1) THEN (Analyze -1)
. . | | | | . |
. . | | | | . 29. RL[(i-1)] = hd(La')
. . | | | | . ⊢  mem_f(S.car;RL[(i-1)];RL)
. . | | | | .   by BackThru
. . | | | | .      Thm* ∀L:T*, i:ℕ||L||. mem_f(T;L[i];L)
. . | | | | .      {select_mem}
. . | | | |
. . | | | .\
. . | | | . 27. ¬j = 0
. . | | | . ⊢
. . | | | . | by (Assert (1≤j)) THEN (Assert (1≤i))
. . | | | . |    THEN
. . | | | . |    (Unfold `select` 0)
. . | | | . |    THEN
. . | | | . |    (RecUnfold `nth_tl` 0)
. . | | | . |    THEN
. . | | | . |    SplitOnConclITE
. . | | | . |    THEN
. . | | | . |    (Auto THEN SplitOnConclITE)
. . | | | . |
. . | | | . .....falsecase .....
```

```
. . | | | . 28. 1≤j
. . | | | . 29. 1≤i
. . | | | . 30. 0<i
. . | | | . 31. 0<j
. . | | | . ⊢  ¬hd(nth_tl(i-1;tl((hd(La').RL))))
. . | | | .      =
. . | | | .      hd(nth_tl(j-1;tl((hd(La').RL))))
. . | | | .   by Reduce 0 THEN Fold `select` 0 THEN BackThru 13
. . | | |
. . | | |\
. . | | | 25. s : S.car
. . | | | 26. hd(La') = s ∨ mem_f(S.car;s;RL)
. . | | | ⊢  ∃w:Alph*. (S:w←si) = s  by Analyze -1
. . | | | |
. . | | | |\
. . | | | | 26. hd(La') = s
. . | | | | ⊢
. . | | | | | by (RWH (RevHypC -1) 0) THEN (Thin -1)
. . | | | | |    THEN
. . | | | | |    (InstHyp [hd(La')] 22)
. . | | | | |
. . | | | | |\
. . | | | | | .....antecedent .....
. . | | | | | 25. S.car
. . | | | | | ⊢  mem_f(S.car;hd(La');La')
. . | | | | |   by BackThru
. . | | | | |      Thm* ∀L:T*. ||L||≥1 ⇒ mem_f(T;hd(L);L)
. . | | | | |      {hd_mem}
. . | | | | |
. . | | | | .\
. . | | | | . 26. mem_f(S.car;hd(La');RLa)
. . | | | | . ⊢  ∃w:Alph*. (S:w←si) = hd(La')  by BackThru 19
. . | | | |
. . | | | .\
. . | | | . 26. mem_f(S.car;s;RL)
. . | | | . ⊢   by BackThru 14
. . | | |
. . | | .\
. . | | . 25. k : ℕ
. . | | . 26. k≤nn
. . | | . ⊢  ∃RLa:S.car*.
. . | | . |  (∀i:{1..(||RL||+1)⁻}, a:Alph.
. . | | . |   hd(La') = S.act(a,(hd(La').RL)[i])
. . | | . |   ∨ mem_f(S.car;S.act(a,(hd(La').RL)[i]);RL)
. . | | . |   ∨ mem_f(S.car;S.act(a,(hd(La').RL)[i]);RLa))
. . | | . |  & (∀a:Alph.
. . | | . |     g(a)<k ⇒
. . | | . |     hd(La') = S.act(a,hd(La'))
. . | | . |     ∨ mem_f(S.car;S.act(a,hd(La'));RL)
. . | | . |     ∨ mem_f(S.car;S.act(a,hd(La'));RLa))
. . | | . |  & (∀s:S.car.
. . | | . |     mem_f(S.car;s;RLa) ⇒ (∃w:Alph*. (S:w←si) = s))
. . | | . | by NatInd -2 THEN Analyze 0
. . | | . |
. . | | . |\
```

```
. . | | . | 25. 0≤nn
. . | | . | ⊢  ∃RLa:S.car*.
. . | | . | |  (∀i:{1..(||RL||+1)⁻}, a:Alph.
. . | | . | |   hd(La') = S.act(a,(hd(La').RL)[i])
. . | | . | |   ∨ mem_f(S.car;S.act(a,(hd(La').RL)[i]);RL)
. . | | . | |   ∨ mem_f(S.car;S.act(a,(hd(La').RL)[i]);RLa))
. . | | . | |  & (∀a:Alph.
. . | | . | |     g(a)<0 ⇒
. . | | . | |     hd(La') = S.act(a,hd(La'))
. . | | . | |     ∨ mem_f(S.car;S.act(a,hd(La'));RL)
. . | | . | |     ∨ mem_f(S.car;S.act(a,hd(La'));RLa))
. . | | . | |  & (∀s:S.car.
. . | | . | |     mem_f(S.car;s;RLa) ⇒ (∃w:Alph*. (S:w←si) = s))
. . | | . | | by (InstHyp [nn] 15) THEN ExRepD
. . | | . | |
. . | | . | 26. R1 : S.car*
. . | | . | 27. ∀i:{1..||RL||⁻}, a:Alph.
. . | | . | |   mem_f(S.car;S.act(a,RL[i]);RL)
. . | | . | |   ∨ mem_f(S.car;S.act(a,RL[i]);R1)
. . | | . | 28. ∀a:Alph.
. . | | . | |   g(a)<nn ⇒
. . | | . | |   mem_f(S.car;S.act(a,hd(RL));RL)
. . | | . | |   ∨ mem_f(S.car;S.act(a,hd(RL));R1)
. . | | . | 29. ∀s:S.car.
. . | | . | |   mem_f(S.car;s;R1) ⇒ (∃w:Alph*. (S:w←si) = s)
. . | | . | ⊢   by InstConcl [R1]
. . | | . | |
. . | | . | |\
. . | | . | | 30. i : {1..(||RL||+1)⁻}
. . | | . | | 31. a : Alph
. . | | . | | ⊢  hd(La') = S.act(a,(hd(La').RL)[i])
. . | | . | | |  ∨ mem_f(S.car;S.act(a,(hd(La').RL)[i]);RL)
. . | | . | | |  ∨ mem_f(S.car;S.act(a,(hd(La').RL)[i]);R1)
. . | | . | | | by Decide (i = 1)
. . | | . | | |
. . | | . | | |\
. . | | . | | | 32. i = 1
. . | | . | | | ⊢
. . | | . | | | | by ((RWH (HypC -1) 0)
. . | | . | | | |     THEN
. . | | . | | | |     (Inst
. . | | . | | | |      Thm* ∀L:T*. 0<||L|| ⇒ L = hd(L).tl(L)
. . | | . | | | |      {hd_tl}
. . | | . | | | |      [S.car;RL])
. . | | . | | | |     THEN
. . | | . | | | |     (RWH (HypC -1) 0)
. . | | . | | | |     THEN
. . | | . | | | |     (Reduce 0))
. . | | . | | | |    THENA
. . | | . | | | |    (Auto THEN Reduce 0)
. . | | . | | | |
. . | | . | | | 33. RL = hd(RL).tl(RL)
. . | | . | | | ⊢  hd(La') = S.act(a,hd(RL))
. . | | . | | | |  ∨ hd(RL) = S.act(a,hd(RL))
. . | | . | | | |  ∨ mem_f(S.car;S.act(a,hd(RL));tl(RL))
```

```
. . | | . | | | |  ∨ mem_f(S.car;S.act(a,hd(RL));R1)
. . | | . | | | | by (InstHyp [a] 28) THEN (Analyze -1)
. . | | . | | | |
. . | | . | | | |\
. . | | . | | | | 34. mem_f(S.car;S.act(a,hd(RL));RL)
. . | | . | | | | ⊢
. . | | . | | | |   by (RWH (HypC -2) -1) THEN (Reduce -1)
. . | | . | | | |      THEN
. . | | . | | | |      (Analyze -1)
. . | | . | | | |      THEN
. . | | . | | | |      ProveProp
. . | | . | | | |
. . | | . | | | .\
. . | | . | | | . 34. mem_f(S.car;S.act(a,hd(RL));R1)
. . | | . | | | . ⊢   by Sel 2 (Analyze 0)
. . | | . | | |
. . | | . | | .\
. . | | . | | . 32. ¬i = 1
. . | | . | | . ⊢
. . | | . | | . | by (Unfold ‘select‘ 0) THEN (RecUnfold ‘nth_tl‘ 0)
. . | | . | | . |    THEN
. . | | . | | . |    SplitOnConclITE
. . | | . | | . |    THEN
. . | | . | | . |    (Try (Fold ‘select‘ 0))
. . | | . | | . |
. . | | . | | . 33. 0<i
. . | | . | | . ⊢  hd(La’) = S.act(a,tl((hd(La’).RL))[(i-1)])
. . | | . | | .    ∨ mem_f(S.car;S.act(a,tl((hd(La’).RL))[(i-1)]);RL)
. . | | . | | .    ∨ mem_f(S.car;S.act(a,tl((hd(La’).RL))[(i-1)]);R1)
. . | | . | | .   by (Reduce 0 THEN InstHyp [i-1;a] 27)
. . | | . | | .      THEN
. . | | . | | .      (Analyze -1)
. . | | . | | .      THEN
. . | | . | | .      ProveProp
. . | | . | |
. . | | . | .\
. . | | . | . 30. s : S.car
. . | | . | . 31. mem_f(S.car;s;R1)
. . | | . | . ⊢  ∃w:Alph*. (S:w←si) = s  by BackThru -3
. . | | . |
. . | | . .\
. . | | . . 25. k : ℤ
. . | | . . 26. 0<k
. . | | . . 27. k-1≤nn ⇒
. . | | . . |   (∃RLa:S.car*.
. . | | . . |    (∀i:{1..(||RL||+1)⁻}, a:Alph.
. . | | . . |     hd(La’) = S.act(a,(hd(La’).RL)[i])
. . | | . . |     ∨ mem_f(S.car;S.act(a,(hd(La’).RL)[i]);RL)
. . | | . . |     ∨ mem_f(S.car;S.act(a,(hd(La’).RL)[i]);RLa))
. . | | . . |    & (∀a:Alph.
. . | | . . |       g(a)<k-1 ⇒
. . | | . . |       hd(La’) = S.act(a,hd(La’))
. . | | . . |       ∨ mem_f(S.car;S.act(a,hd(La’));RL)
. . | | . . |       ∨ mem_f(S.car;S.act(a,hd(La’));RLa))
. . | | . . |    & (∀s:S.car.
```

```
. . | | . . |        mem_f(S.car;s;RLa) ⇒ (∃w:Alph*. (S:w←si) = s)))
. . | | . . 28. k≤nn
. . | | . . ⊢   by (Analyze -2) THEN ExRepD
. . | | . . |
. . | | . . 27. k≤nn
. . | | . . 28. R1 : S.car*
. . | | . . 29. ∀i:{1..(||RL||+1)⁻}, a:Alph.
. . | | . . |   hd(La') = S.act(a,(hd(La').RL)[i])
. . | | . . |   ∨ mem_f(S.car;S.act(a,(hd(La').RL)[i]);RL)
. . | | . . |   ∨ mem_f(S.car;S.act(a,(hd(La').RL)[i]);R1)
. . | | . . 30. ∀a:Alph.
. . | | . . |   g(a)<k-1 ⇒
. . | | . . |   hd(La') = S.act(a,hd(La'))
. . | | . . |   ∨ mem_f(S.car;S.act(a,hd(La'));RL)
. . | | . . |   ∨ mem_f(S.car;S.act(a,hd(La'));R1)
. . | | . . 31. ∀s:S.car.
. . | | . . |   mem_f(S.car;s;R1) ⇒ (∃w:Alph*. (S:w←si) = s)
. . | | . . ⊢
. . | | . . | by (InstConcl [S.act(f(k-1),hd(La')).R1])
. . | | . . |    THEN
. . | | . . |    (Reduce 0)
. . | | . . |
. . | | . . |\
. . | | . . | 32. i : {1..(||RL||+1)⁻}
. . | | . . | 33. a : Alph
. . | | . . | ⊢  hd(La') = S.act(a,(hd(La').RL)[i])
. . | | . . |    ∨ mem_f(S.car;S.act(a,(hd(La').RL)[i]);RL)
. . | | . . |    ∨ S.act(f(k-1),hd(La')) = S.act(a,(hd(La').RL)[i])
. . | | . . |    ∨ mem_f(S.car;S.act(a,(hd(La').RL)[i]);R1)
. . | | . . |   by (InstHyp [i;a] 29) THEN (Analyze -1)
. . | | . . |      THEN
. . | | . . |      ProveProp

. . | | . . |
. . | | . . |\
. . | | . . | 32. a : Alph
. . | | . . | 33. g(a)<k
. . | | . . | ⊢  hd(La') = S.act(a,hd(La'))
. . | | . . | |  ∨ mem_f(S.car;S.act(a,hd(La'));RL)
. . | | . . | |  ∨ S.act(f(k-1),hd(La')) = S.act(a,hd(La'))
. . | | . . | |  ∨ mem_f(S.car;S.act(a,hd(La'));R1)
. . | | . . | | by Decide (g(a) = k-1)
. . | | . . | |
. . | | . . | |\
. . | | . . | | 34. g(a) = k-1
. . | | . . | | ⊢   by (Sel 2 (Analyze 0)) THEN (RWH (RevHypC -1) 0)
. . | | . . | | |
. . | | . . | | ⊢  S.act(f(g(a)),hd(La')) = S.act(a,hd(La'))
. . | | . . | |   by (RWH (AddrC [2;1;2] add_composeC) 0)
. . | | . . | |      THEN
. . | | . . | |      (Analyze 8)
. . | | . . | |      THEN
. . | | . . | |      (RWH (HypC 9) 0)
. . | | . . | |      THEN
. . | | . . | |      (Reduce 0)
. . | | . . | |
```

```
. . | | . . | .\
. . | | . . | . 34. ¬g(a) = k-1
. . | | . . | . ⊢   by (InstHyp [a] 30) THEN ProveProp
. . | | . . |
. . | | . . .\
. . | | . . . 32. s : S.car
. . | | . . . 33. S.act(f(k-1),hd(La')) = s ∨ mem_f(S.car;s;R1)
. . | | . . . ⊢  ∃w:Alph*. (S:w←si) = s  by Analyze -1
. . | | . . . |
. . | | . . . |\
. . | | . . . | 33. S.act(f(k-1),hd(La')) = s
. . | | . . . | ⊢
. . | | . . . | | by (Inst
. . | | . . . | |     Thm* ∀L:T*. ||L||≥1 ⇒ mem_f(T;hd(L);L)
. . | | . . . | |     {hd_mem}
. . | | . . . | |     [S.car;La'])
. . | | . . . | |    THEN
. . | | . . . | |    (FwdThru 22 [-1])
. . | | . . . | |    THEN
. . | | . . . | |    (FwdThru 19 [-1])
. . | | . . . | |
. . | | . . . | 34. mem_f(S.car;hd(La');La')
. . | | . . . | 35. mem_f(S.car;hd(La');RLa)
. . | | . . . | 36. ∃w:Alph*. (S:w←si) = hd(La')
. . | | . . . | ⊢
. . | | . . . | | by (Analyze -1 THEN InstConcl [f(k-1).w])
. . | | . . . | |    THEN
. . | | . . . | |    (RWH (RevHypC 33) 0)
. . | | . . . | |
. . | | . . . | 36. w : Alph*
. . | | . . . | 37. (S:w←si) = hd(La')
. . | | . . . | ⊢  (S:f(k-1).w←si) = S.act(f(k-1),hd(La'))
. . | | . . . |   by RecUnfold `maction` 0 THEN Reduce 0
. . | | . . . |      THEN
. . | | . . . |      RWH (HypC -1) 0
. . | | . . . |
. . | | . . . .\
. . | | . . . . 33. mem_f(S.car;s;R1)
. . | | . . . . ⊢   by BackThru -3
. . | |
. . | |\
. . | | 25. R1 :
. . | |     {y:{x:(S.car*)| 0<||x|| & ||x||≤n+1 }| y[(||y||-1)] = si }
. . | | 26. k : ℕ
. . | | 27. k≤nn
. . | | 28. S.car*
. . | | 29. a : Alph
. . | | 30. g(a)<k
. . | | ⊢  ||R1||≥1
. . | |   by Analyze 25 THEN Analyze 25 THEN UnhideSqStableHyp 27
. . | |
. . | .\
. . | . 25. R1 :
. . | .     {y:{x:(S.car*)| 0<||x|| & ||x||≤n+1 }| y[(||y||-1)] = si }
. . | . 26. k : ℕ
```

```
. . | . 27. k≤nn
. . | . 28. S.car*
. . | . 29. a : Alph
. . | . 30. g(a)<k
. . | . ⊢  ||R1||≥1
. . | .   by Analyze 25 THEN Analyze 25 THEN UnhideSqStableHyp 27
. . |
. . .\
. . . 24. ¬||La'||≥1
. . . ⊢
. . . | by (Inst Thm* ∀L:T*. ||L||≤0 ⇒ L = nil    {zero_nil}
. . . |      [S.car;La'])
. . . |    THEN
. . . |    (Thin 24)
. . . |    THEN
. . . |    (Thin 23)
. . . |
. . . 23. La' = nil
. . . ⊢   by (InstConcl [RL]) THEN (Sel 1 (Analyze 0)) THEN GenRepD
. . . |
. . . |\
. . . | 11. RL : S.car*
. . . | 12. 0<||RL||
. . . | 13. ||RL||≤n-1+1
. . . | 14. RL[(||RL||-1)] = si
. . . | 15. ||RL|| = n-1+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. ∀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)))
. . . | 19. RLa : S.car*
. . . | 20. ∀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)
. . . | 21. ∀a:Alph.
. . . |     g(a)<nn ⇒
. . . |     mem_f(S.car;S.act(a,hd(RL));RL)
. . . |     ∨ mem_f(S.car;S.act(a,hd(RL));RLa)
. . . | 22. ∀s:S.car. mem_f(S.car;s;RLa) ⇒ (∃w:Alph*. (S:w←si) = s)
. . . | 23. La' : S.car*
. . . | 24. ∀t:S.car.
. . . |     mem_f(S.car;t;RLa) ⇒
. . . |     mem_f(S.car;t;RL) ∨ mem_f(S.car;t;La')
. . . | 25. ∀t:S.car. mem_f(S.car;t;La') ⇒ mem_f(S.car;t;RLa)
. . . | 26. La' = nil
```

```
. . . | 27. y : {x:(S.car*)| 0<||x|| & ||x||≤n+1 }
. . . | ⊢  0≤||y||-1  by Analyze -1
. . . |
. . . |\
. . . | 24. s : S.car
. . . | 25. ∃w:Alph*. (S:w←si) = s
. . . | ⊢  mem_f(S.car;s;RL)
. . . | | by Assert (∀w:Alph*. mem_f(S.car;(S:w←si);RL))
. . . | |
. . . | |\
. . . | | .....assertion .....
. . . | | ⊢  ∀w:Alph*. mem_f(S.car;(S:w←si);RL)
. . . | | | by (Thin -1 THEN RecUnfold ‘maction‘ 0 THEN Analyze 0)
. . . | | |    THEN
. . . | | |    (ListInd -1)
. . . | | |    THEN
. . . | | |    (Reduce 0)
. . . | | |
. . . | | |\
. . . | | | 25. Alph*
. . . | | | ⊢  mem_f(S.car;si;RL)
. . . | | |   by (Analyze 11 THEN HUnhide) THEN (RWH (RevHypC 12) 0)
. . . | | |      THEN
. . . | | |      (BackThru
. . . | | |       Thm* ∀L:T*, i:ℕ||L||. mem_f(T;L[i];L)
. . . | | |       {select_mem})
. . . | | |
. . . | | .\
. . . | | . 25. Alph*
. . . | | . 26. u : Alph
. . . | | . 27. v : Alph*
. . . | | . 28. mem_f(S.car;if null(v)→ si
. . . | | . |                else S.act(hd(v),(S:tl(v)←si)) fi;RL)
. . . | | . ⊢  mem_f(S.car;S.act(u,(S:v←si));RL)
. . . | | . | by (RecFold ‘maction‘ -1)
. . . | | . |    THEN
. . . | | . |    (Inst
. . . | | . |     Thm* ∀L:T*, t:T.
. . . | | . |          mem_f(T;t;L) ⇒ (∃i:ℕ||L||. L[i] = t)
. . . | | . |     {mem_select}
. . . | | . |     [S.car;RL;(S:v←si)])
. . . | | . |    THEN
. . . | | . |    (Analyze -1)
. . . | | . |
. . . | | . 28. mem_f(S.car;(S:v←si);RL)
. . . | | . 29. i : ℕ||RL||
. . . | | . 30. RL[i] = (S:v←si)
. . . | | . ⊢   by Decide (i = 0)
. . . | | . |
. . . | | . |\
. . . | | . | 31. i = 0
. . . | | . | ⊢
. . . | | . | | by (RWH (HypC -1) -2) THEN (Unfold ‘select‘ -2)
. . . | | . | |    THEN
. . . | | . | |    (Reduce -2)
```

```
. . . | | . | |
. . . | | . | 30. hd(RL) = (S:v←si)
. . . | | . | 31. i = 0
. . . | | . | ⊢
. . . | | . | | by (RWH (RevHypC -2) 0) THEN (InstHyp [u] 18)
. . . | | . | |    THEN
. . . | | . | |    (Analyze -1)
. . . | | . | |
. . . | | . | 32. mem_f(S.car;S.act(u,hd(RL));RLa)
. . . | | . | ⊢  mem_f(S.car;S.act(u,hd(RL));RL)
. . . | | . | | by (FwdThru 21 [-1]) THEN (Analyze -1)
. . . | | . | |
. . . | | . | 33. mem_f(S.car;S.act(u,hd(RL));La')
. . . | | . | ⊢   by (RWH (HypC 23) -1) THEN (Reduce -1)
. . . | | . |
. . . | | . .\
. . . | | . . 31. ¬i = 0
. . . | | . . ⊢   by (InstHyp [i;u] 17) THEN (Analyze -1)
. . . | | . . |
. . . | | . . |\
. . . | | . . | 32. mem_f(S.car;S.act(u,RL[i]);RL)
. . . | | . . | ⊢   by RWH (HypC -3) -1
. . . | | . . |
. . . | | . . .\
. . . | | . . . 32. mem_f(S.car;S.act(u,RL[i]);RLa)
. . . | | . . . ⊢
. . . | | . . . | by (RWH (RevHypC -3) 0) THEN (FwdThru 21 [-1])
. . . | | . . . |    THEN
. . . | | . . . |    (Analyze -1)
. . . | | . . . |
. . . | | . . . 33. mem_f(S.car;S.act(u,RL[i]);La')
. . . | | . . . ⊢  mem_f(S.car;S.act(u,RL[i]);RL)
. . . | | . . .   by (RWH (HypC 23) -1) THEN (Reduce -1)
. . . | |
. . . | .\
. . . | . 26. ∀w:Alph*. mem_f(S.car;(S:w←si);RL)
. . . | . ⊢   by Analyze -2 THEN InstHyp [w] -1
. . . | . |
. . . | . 25. w : Alph*
. . . | . 26. (S:w←si) = s
. . . | . 27. ∀w:Alph*. mem_f(S.car;(S:w←si);RL)
. . . | . 28. mem_f(S.car;(S:w←si);RL)
. . . | . ⊢   by RWH (HypC -3) -1
. . . |
. . . |\
. . . | 24. s : S.car
. . . | 25. mem_f(S.car;s;RL)
. . . | ⊢  ∃w:Alph*. (S:w←si) = s  by BackThru 14
. . . |
. . . |\
. . . | 24. R1 :
. . . |     {y:{x:(S.car*)| 0<||x|| & ||x||≤n+1 }| y[(||y||-1)] = si }
. . . | 25. k : ℕ
. . . | 26. k≤nn
. . . | 27. S.car*
```

```
. . . | 28. a : Alph
. . . | 29. g(a)<k
. . . | ⊢  ||R1||≥1
. . . |   by Analyze 24 THEN Analyze 24 THEN UnhideSqStableHyp 26
. . . |
. . . .\
. . . . 24. R1 :
. . . .     {y:{x:(S.car*)| 0<||x|| & ||x||≤n+1 }| y[(||y||-1)] = si }
. . . . 25. k : ℕ
. . . . 26. k≤nn
. . . . 27. S.car*
. . . . 28. a : Alph
. . . . 29. g(a)<k
. . . . ⊢  ||R1||≥1
. . . .   by Analyze 24 THEN Analyze 24 THEN UnhideSqStableHyp 26
END
```