```
back_listify  THM

⊢  ∀S:ActionSet(Alph), s:S.car.
|  Fin(Alph) ⇒
|  Fin(S.car) ⇒
|  (∃BL:S.car*.
|   ∀t:S.car. mem_f(S.car;t;BL) ⟺ (∃a:Alph. S.act(a,t) = s))
| by RepD
|
1. Alph : Type
2. S : ActionSet(Alph)
3. s : S.car
4. Fin(Alph)
5. Fin(S.car)
⊢  ∃BL:S.car*.
|  ∀t:S.car. mem_f(S.car;t;BL) ⟺ (∃a:Alph. S.act(a,t) = s)
| by Assert
|    (∀LL:(S.car×S.car*)*.
|     ∃BL:S.car*.
|     ∀t:S.car.
|     mem_f(S.car;t;BL)
|     ⟺
|     (∃i:ℕ||LL||. 1of(LL[i]) = t & mem_f(S.car;s;2of(LL[i]))))
|
|\
| .....assertion .....
| ⊢  ∀LL:(S.car×S.car*)*.
| |  ∃BL:S.car*.
| |  ∀t:S.car.
| |  mem_f(S.car;t;BL)
| |  ⟺
| |  (∃i:ℕ||LL||. 1of(LL[i]) = t & mem_f(S.car;s;2of(LL[i])))
| | by (Analyze 0) THEN (ListInd -1)
| |
| |\
| | 6. (S.car×S.car*)*
| | ⊢  ∃BL:S.car*.
| | |  ∀t:S.car.
| | |  mem_f(S.car;t;BL)
| | |  ⟺
| | |  (∃i:ℕ||nil||. 1of(nil[i]) = t & mem_f(S.car;s;2of(nil[i])))
| | | by (Reduce 0 THEN InstConcl [nil]) THEN (Reduce 0)
| | |
| | 7. t : S.car
| | 8. ∃i:ℕ0. 1of(nil[i]) = t & mem_f(S.car;s;2of(nil[i]))
| | ⊢  False  by Analyze -1
| |
| .\
| . 6. (S.car×S.car*)*
| . 7. u : S.car×S.car*
| . 8. v : (S.car×S.car*)*
| . 9. ∃BL:S.car*.
| . |  ∀t:S.car.
| . |  mem_f(S.car;t;BL)
| . |  ⟺
```

```
| . |  (∃i:ℕ||v||. 1of(v[i]) = t & mem_f(S.car;s;2of(v[i])))
| . ⊢  ∃BL:S.car*.
| . |  ∀t:S.car.
| . |  mem_f(S.car;t;BL)
| . |  ⟺
| . |  (∃i:ℕ||u.v||. 1of((u.v)[i]) = t & mem_f(S.car;s;2of((u.v)[i])))
| . | by Analyze -1 THEN Analyze 7
| . |
| . 7. u1 : S.car
| . 8. u2 : S.car*
| . 9. v : (S.car×S.car*)*
| . 10. BL : S.car*
| . 11. ∀t:S.car.
| . |   mem_f(S.car;t;BL)
| . |   ⟺
| . |   (∃i:ℕ||v||. 1of(v[i]) = t & mem_f(S.car;s;2of(v[i])))
| . ⊢  ∃BL:S.car*.
| . |  ∀t:S.car.
| . |  mem_f(S.car;t;BL)
| . |  ⟺
| . |  (∃i:ℕ||<u1,u2>.v||.
| . |   1of((<u1,u2>.v)[i]) = t & mem_f(S.car;s;2of((<u1,u2>.v)[i])))
| . | by (Inst
| . |     Thm* ∀s:S, l:S*. Fin(S) ⇒ Dec(mem_f(S;s;l))    {mem_f_dec}
| . |     [S.car;s;u2])
| . |    THEN
| . |    (Analyze -1)
| . |
| . |\
| . | 12. mem_f(S.car;s;u2)
| . | ⊢   by (InstConcl [u1.BL]) THEN (Reduce 0)
| . | |
| . | |\
| . | | 13. t : S.car
| . | | 14. u1 = t ∨ mem_f(S.car;t;BL)
| . | | ⊢  ∃i:ℕ(||v||+1).
| . | | |  1of((<u1,u2>.v)[i]) = t
| . | | |  & mem_f(S.car;s;2of((<u1,u2>.v)[i]))  by Analyze -1
| . | | |
| . | | |\
| . | | | 14. u1 = t
| . | | | ⊢   by (InstConcl [0]) THEN (Reduce 0)
| . | | |
| . | | .\
| . | | . 14. mem_f(S.car;t;BL)
| . | | . ⊢   by (FwdThru 11 [-1]) THEN ExRepD
| . | | . |
| . | | . 15. i : ℕ||v||
| . | | . 16. 1of(v[i]) = t
| . | | . 17. mem_f(S.car;s;2of(v[i]))
| . | | . ⊢
| . | | . | by (InstConcl [i+1]) THEN (Unfold ‘select‘ 0)
| . | | . |    THEN
| . | | . |    (RecUnfold ‘nth_tl‘ 0)
| . | | . |    THEN
```

```
| . | | . |    SplitOnConclITE
| . | | . |    THEN
| . | | . |    (Reduce 0)
| . | | . |
| . | | . |\
| . | | . | 18. 0<i+1
| . | | . | ⊢  1of(hd(nth_tl(i+1-1;v))) = t
| . | | . |   by Fold ‘select‘ 0 THEN ArithSimp 0
| . | | . |
| . | | . .\
| . | | . . 18. 0<i+1
| . | | . . ⊢  mem_f(S.car;s;2of(hd(nth_tl(i+1-1;v))))
| . | | . .   by Fold ‘select‘ 0 THEN ArithSimp 0
| . | |
| . | .\
| . | . 13. t : S.car
| . | . 14. ∃i:ℕ(||v||+1).
| . | . |   1of((<u1,u2>.v)[i]) = t
| . | . |   & mem_f(S.car;s;2of((<u1,u2>.v)[i]))
| . | . ⊢  u1 = t ∨ mem_f(S.car;t;BL)
| . | . | by Analyze -1 THEN Decide (i = 0)
| . | . |
| . | . |\
| . | . | 14. i : ℕ(||v||+1)
| . | . | 15. 1of((<u1,u2>.v)[i]) = t
| . | . |     & mem_f(S.car;s;2of((<u1,u2>.v)[i]))
| . | . | 16. i = 0
| . | . | ⊢   by (RWH (HypC -1) -2) THEN (Reduce -2) THEN ProveProp
| . | . |
| . | . .\
| . | . . 14. i : ℕ(||v||+1)
| . | . . 15. 1of((<u1,u2>.v)[i]) = t
| . | . . |   & mem_f(S.car;s;2of((<u1,u2>.v)[i]))
| . | . . 16. ¬i = 0
| . | . . ⊢
| . | . . | by (Unfold ‘select‘ -2 THEN RecUnfold ‘nth_tl‘ -2
| . | . . |     THEN
| . | . . |     MoveToConcl -2
| . | . . |     THEN
| . | . . |     SplitOnConclITE)
| . | . . |    THEN
| . | . . |    (Reduce 0)
| . | . . |    THEN
| . | . . |    (Try (Fold ‘select‘ 0))
| . | . . |
| . | . . 15. ¬i = 0
| . | . . 16. 0<i
| . | . . 17. 1of(v[(i-1)]) = t
| . | . . 18. mem_f(S.car;s;2of(v[(i-1)]))
| . | . . ⊢   by (Sel 2 (Analyze 0)) THEN (BackThru 11)
| . | . . |
| . | . . ⊢  ∃i:ℕ||v||. 1of(v[i]) = t & mem_f(S.car;s;2of(v[i]))
| . | . .   by InstConcl [i-1]
| . |
| . .\
```

```
| . . 12. ¬mem_f(S.car;s;u2)
| . . ⊢   by InstConcl [BL]
| . . |
| . . |\
| . . | 13. t : S.car
| . . | 14. mem_f(S.car;t;BL)
| . . | ⊢  ∃i:ℕ||<u1,u2>.v||.
| . . | |  1of((<u1,u2>.v)[i]) = t
| . . | |  & mem_f(S.car;s;2of((<u1,u2>.v)[i]))
| . . | | by (FwdThru 11 [14]) THEN (Analyze -1)
| . . | |    THEN
| . . | |    (InstConcl [i+1])
| . . | |
| . . | 15. i : ℕ||v||
| . . | 16. 1of(v[i]) = t & mem_f(S.car;s;2of(v[i]))
| . . | ⊢  1of((<u1,u2>.v)[(i+1)]) = t
| . . |    & mem_f(S.car;s;2of((<u1,u2>.v)[(i+1)]))
| . . |   by (Unfold `select` 0 THEN RecUnfold `nth_tl` 0
| . . |       THEN
| . . |       SplitOnConclITE)
| . . |      THEN
| . . |      (Reduce 0)
| . . |      THEN
| . . |      (Try (Fold `select` 0 THEN ArithSimp 0))
| . . |
| . . .\
| . . . 13. t : S.car
| . . . 14. ∃i:ℕ||<u1,u2>.v||.
| . . . |   1of((<u1,u2>.v)[i]) = t
| . . . |   & mem_f(S.car;s;2of((<u1,u2>.v)[i]))
| . . . ⊢  mem_f(S.car;t;BL)
| . . . | by (Analyze -1 THEN MoveToConcl -1 THEN Unfold `select` 0
| . . . |     THEN
| . . . |     RecUnfold `nth_tl` 0
| . . . |     THEN
| . . . |     SplitOnConclITE)
| . . . |    THEN
| . . . |    (Reduce 0)
| . . . |
| . . . 14. i : ℕ||<u1,u2>.v||
| . . . 15. 0<i
| . . . 16. 1of(hd(nth_tl(i-1;v))) = t
| . . . 17. mem_f(S.car;s;2of(hd(nth_tl(i-1;v))))
| . . . ⊢
| . . . | by Fold `select` -1 THEN Fold `select` -2 THEN BackThru 11
| . . . |
| . . . 16. 1of(v[(i-1)]) = t
| . . . 17. mem_f(S.car;s;2of(v[(i-1)]))
| . . . ⊢  ∃i:ℕ||v||. 1of(v[i]) = t & mem_f(S.car;s;2of(v[i]))
| . . .   by InstConcl [i-1]
|
.\
. 6. ∀LL:(S.car×S.car*)*.
. |  ∃BL:S.car*.
. |  ∀t:S.car.
```

```
. |  mem_f(S.car;t;BL)
. |  ⟺
. |  (∃i:ℕ||LL||. 1of(LL[i]) = t & mem_f(S.car;s;2of(LL[i])))
. ⊢   by Analyze -3 THEN Analyze -3 THEN Analyze -2 THEN Analyze -2
. |
. 4. n : ℕ
. 5. f : ℕn→Alph
. 6. Bij(ℕn; Alph; f)
. 7. n1 : ℕ
. 8. f1 : ℕn1→S.car
. 9. Bij(ℕn1; S.car; f1)
. 10. ∀LL:(S.car×S.car*)*.
. |   ∃BL:S.car*.
. |   ∀t:S.car.
. |   mem_f(S.car;t;BL)
. |   ⟺
. |   (∃i:ℕ||LL||. 1of(LL[i]) = t & mem_f(S.car;s;2of(LL[i])))
. ⊢
. | by (InstHyp [(λx.<f1(x),(λy.S.act(f(y),f1(x)))[ℕn]>)[ℕn1]] -1)
. |    THEN
. |    (Thin -2)
. |
. 10. ∃BL:S.car*.
. |   ∀t:S.car.
. |   mem_f(S.car;t;BL)
. |   ⟺
. |   (∃i:ℕ||(λx.<f1(x),(λy.S.act(f(y),f1(x)))[ℕn]>)[ℕn1]||.
. |    1of(((λx.<f1(x),(λy.S.act(f(y),f1(x)))[ℕn]>)[ℕn1])[i]) = t
. |    & mem_f(S.car;s;2of(((λx.<f1(x)
. |                               ,(λy.S.act
. |                                    (f(y)
. |                                    ,f1(x)))[ℕn]>)[ℕn1])[i])))
. ⊢
. | by (RWH
. |     (LemmaC
. |      Thm* ∀n:ℕ, f:(ℕn→T), i:ℕn. ((f)[ℕn])[i] = f(i)
. |      {select_listify_id})
. |     -1)
. |    THEN
. |    (Reduce -1)
. |    THEN
. |    (Analyze -1)
. |
. 10. BL : S.car*
. 11. ∀t:S.car.
. |   mem_f(S.car;t;BL)
. |   ⟺
. |   (∃i:ℕ||(λx.<f1(x),(λy.S.act(f(y),f1(x)))[ℕn]>)[ℕn1]||.
. |    f1(i) = t & mem_f(S.car;s;(λy.S.act(f(y),f1(i)))[ℕn]))
. ⊢   by (InstConcl [BL] THEN InstHyp [t] 11) THEN (Thin 11)
. |
. |\
. | 11. t : S.car
. | 12. mem_f(S.car;t;BL)
. | 13. mem_f(S.car;t;BL) ⇒
```

```
. | |   (∃i:ℕ||(λx.<f1(x),(λy.S.act(f(y),f1(x)))[ℕn]>)[ℕn1]||.
. | |    f1(i) = t & mem_f(S.car;s;(λy.S.act(f(y),f1(i)))[ℕn]))
. | 14. mem_f(S.car;t;BL) ⇐ (∃i:ℕ||(λx.<f1(x)
. | |                                 ,(λy.S.act
. | |                                      (f(y)
. | |                                      ,f1(x)))[ℕn]>)[ℕn1]||.
. | |   f1(i) = t & mem_f(S.car;s;(λy.S.act(f(y),f1(i)))[ℕn]))
. | ⊢  ∃a:Alph. S.act(a,t) = s
. | | by FwdThru 13 [12] THEN RepeatFor 3 (Thin -2) THEN ExRepD
. | |
. | 12. i : ℕ||(λx.<f1(x),(λy.S.act(f(y),f1(x)))[ℕn]>)[ℕn1]||
. | 13. f1(i) = t
. | 14. mem_f(S.car;s;(λy.S.act(f(y),f1(i)))[ℕn])
. | ⊢
. | | by FwdThru
. | |    Thm* ∀L:T*, t:T. mem_f(T;t;L) ⇒ (∃i:ℕ||L||. L[i] = t)
. | |    {mem_select}
. | |    [-1]
. | |
. | 15. ∃i1:ℕ||(λy.S.act(f(y),f1(i)))[ℕn]||.
. | |   ((λy.S.act(f(y),f1(i)))[ℕn])[i1] = s
. | ⊢
. | | by (RWH
. | |     (LemmaC
. | |      Thm* ∀n:ℕ, f:(ℕn→T), i:ℕn. ((f)[ℕn])[i] = f(i)
. | |      {select_listify_id})
. | |     -1)
. | |    THEN
. | |    (Reduce -1)
. | |    THEN
. | |    (Analyze -1)
. | |
. | 15. i1 : ℕ||(λy.S.act(f(y),f1(i)))[ℕn]||
. | 16. S.act(f(i1),f1(i)) = s
. | ⊢   by (InstConcl [f(i1)]) THEN (RWH (HypC 13) -1)
. |
. .\
. . 11. t : S.car
. . 12. ∃a:Alph. S.act(a,t) = s
. . 13. mem_f(S.car;t;BL) ⇒
. . |   (∃i:ℕ||(λx.<f1(x),(λy.S.act(f(y),f1(x)))[ℕn]>)[ℕn1]||.
. . |    f1(i) = t & mem_f(S.car;s;(λy.S.act(f(y),f1(i)))[ℕn]))
. . 14. mem_f(S.car;t;BL) ⇐ (∃i:ℕ||(λx.<f1(x)
. . |                                 ,(λy.S.act
. . |                                      (f(y)
. . |                                      ,f1(x)))[ℕn]>)[ℕn1]||.
. . |   f1(i) = t & mem_f(S.car;s;(λy.S.act(f(y),f1(i)))[ℕn]))
. . ⊢  mem_f(S.car;t;BL)
. . | by BackThru -1 THEN Thin -1 THEN Thin -1 THEN Analyze -1
. . |
. . 12. a : Alph
. . 13. S.act(a,t) = s
. . ⊢  ∃i:ℕ||(λx.<f1(x),(λy.S.act(f(y),f1(x)))[ℕn]>)[ℕn1]||.
. . |  f1(i) = t & mem_f(S.car;s;(λy.S.act(f(y),f1(i)))[ℕn])
. . | by (Analyze 9 THEN Thin 9 THEN Unfold ‘surject‘ 9
```

```
. . |     THEN
. . |     InstHyp [t] 9)
. . |    THEN
. . |    (Analyze -1)
. . |
. . 9. ∀b:S.car. ∃a:ℕn1. f1(a) = b
. . 10. S.car*
. . 11. t : S.car
. . 12. a : Alph
. . 13. S.act(a,t) = s
. . 14. a1 : ℕn1
. . 15. f1(a1) = t
. . ⊢   by (InstConcl [a1]) THEN (RWH (HypC -1) 0)
. . |
. . ⊢  mem_f(S.car;s;(λy.S.act(f(y),t))[ℕn])
. . | by (Thin -1 THEN Thin -1 THEN Analyze 6 THEN Thin 6
. . |     THEN
. . |     Unfold ‘surject‘ 6
. . |     THEN
. . |     InstHyp [a] 6)
. . |    THEN
. . |    (Analyze -1)
. . |
. . 6. ∀b:Alph. ∃a:ℕn. f(a) = b
. . 7. n1 : ℕ
. . 8. f1 : ℕn1→S.car
. . 9. ∀b:S.car. ∃a:ℕn1. f1(a) = b
. . 10. S.car*
. . 11. t : S.car
. . 12. a : Alph
. . 13. S.act(a,t) = s
. . 14. a@0 : ℕn
. . 15. f(a@0) = a
. . ⊢
. . | by Inst Thm* ∀L:T*, i:ℕ||L||. mem_f(T;L[i];L)    {select_mem}
. . |    [S.car;(λy.S.act(f(y),t))[ℕn];a@0]
. . |
. . 16. mem_f(S.car;((λy.S.act(f(y),t))[ℕn])[a@0];(λy.S.act
. .                                                      (f(y)
. .                                                      ,t))[ℕn])
. . ⊢
. .   by (RWH
. .       (LemmaC
. .        Thm* ∀n:ℕ, f:(ℕn→T), i:ℕn. ((f)[ℕn])[i] = f(i)
. .        {select_listify_id})
. .       -1)
. .      THEN
. .      (Reduce -1)
. .      THEN
. .      (RWH (HypC -2) -1)
. .      THEN
. .      (RWH (HypC 13) -1)
END
```