```
pump_theorem  THM

⊢  ∀S:ActionSet(Alph), N:ℕ+, s:S.car.
|  #(S.car)=N  ⇒
|  (∀A:Alph*.
|   N<||A|| ⇒
|   (∃a,b,c:Alph*.
|    0<||b|| & A = ((a @ b) @ c)
|    & (∀k:ℕ. (S:(a @ (b↑k)) @ c←s) = (S:A←s))))  by GenUnivCD
|
1. Alph : Type
2. S : ActionSet(Alph)
3. N : ℕ+
4. s : S.car
5. #(S.car)=N
6. A : Alph*
7. N<||A||
⊢  ∃a,b,c:Alph*.
|  0<||b|| & A = ((a @ b) @ c)
|  & (∀k:ℕ. (S:(a @ (b↑k)) @ c←s) = (S:A←s))  by Unfold `card` 5
|
5. S.car ~ ℕN
6. A : Alph*
7. N<||A||
⊢   by Unfold `one_one_corr` 5
|
5. ∃f:(S.car→ℕN), g:(ℕN→S.car). InvFuns(S.car; ℕN; f; g)
6. A : Alph*
7. N<||A||
⊢   by Analyze 5
|
5. f : S.car→ℕN
6. ∃g:(ℕN→S.car). InvFuns(S.car; ℕN; f; g)
7. A : Alph*
8. N<||A||
⊢
| by Inst
|    Thm* ∀n:{1...}, m:{n+1...}, f:(ℕm→ℕn).
|         ∃i,j:ℕm. i<j & f(i) = f(j)
|    {phole_lemma}
|    [N;||A||;λi.f((S:A[||A||-i..||A||⁻]←s))]
|
9. ∃i,j:ℕ||A||.
|  i<j
|  & (λi.f((S:A[||A||-i..||A||⁻]←s)))(i)
|    =
|    (λi.f((S:A[||A||-i..||A||⁻]←s)))(j)
⊢   by Analyze 9
|
9. i : ℕ||A||
10. ∃j:ℕ||A||.
|   i<j
|   & (λi.f((S:A[||A||-i..||A||⁻]←s)))(i)
|     =
|     (λi.f((S:A[||A||-i..||A||⁻]←s)))(j)
```

```
⊢   by Analyze 10
|
10. j : ℕ||A||
11. i<j
|   & (λi.f((S:A[||A||-i..||A||⁻]←s)))(i)
|     =
|     (λi.f((S:A[||A||-i..||A||⁻]←s)))(j)
⊢   by Analyze 11
|
11. i<j
12. (λi.f((S:A[||A||-i..||A||⁻]←s)))(i)
|    =
|    (λi.f((S:A[||A||-i..||A||⁻]←s)))(j)
⊢   by Witness A[0..||A||-j⁻]
|
⊢  ∃b,c:Alph*.
|  0<||b|| & A = (((A[0..||A||-j⁻]) @ b) @ c)
|  & (∀k:ℕ. (S:((A[0..||A||-j⁻]) @ (b↑k)) @ c←s) = (S:A←s))
| by Witness A[||A||-j..||A||-i⁻]
|
⊢  ∃c:Alph*.
|  0<||A[||A||-j..||A||-i⁻]||
|  & A = (((A[0..||A||-j⁻]) @ (A[||A||-j..||A||-i⁻])) @ c)
|  & (∀k:ℕ.
|     (S:((A[0..||A||-j⁻]) @ (A[||A||-j..||A||-i⁻]↑k)) @ c←s)
|     =
|     (S:A←s))  by Witness A[||A||-i..||A||⁻]
|
⊢  0<||A[||A||-j..||A||-i⁻]||
|  & A
|    =
|    (((A[0..||A||-j⁻]) @ (A[||A||-j..||A||-i⁻]))
|     @ (A[||A||-i..||A||⁻]))
|  & (∀k:ℕ.
|     (S:((A[0..||A||-j⁻]) @ (A[||A||-j..||A||-i⁻]↑k))
|     @ (A[||A||-i..||A||⁻])←s)
|     =
|     (S:A←s))  by Analyze 0
|
|\
| ⊢  0<||A[||A||-j..||A||-i⁻]||
| |  & A
| |    =
| |    (((A[0..||A||-j⁻]) @ (A[||A||-j..||A||-i⁻]))
| |     @ (A[||A||-i..||A||⁻]))  by Analyze 0
| |
| |\
| | ⊢  0<||A[||A||-j..||A||-i⁻]||
| |   by Inst
| |      Thm* ∀as:T*, i:{0...||as||}, j:{i...||as||}.
| |           ||as[i..j⁻]|| = j-i
| |      {length_segment}
| |      [Alph;A;||A||-j;||A||-i]
| |      THEN
| |      SupInf
```

```
| |
| .\
| . ⊢  A
| . |  =
| . |  (((A[0..||A||-j⁻]) @ (A[||A||-j..||A||-i⁻]))
| . |   @ (A[||A||-i..||A||⁻]))
| . | by RW
| . |    (SweepUpC
| . |     (LemmaC
| . |      Thm* ∀as:T*, i:{0...||as||}, j:{i...||as||}
| . |           , k:{j...||as||}.
| . |           ((as[i..j⁻]) @ (as[j..k⁻])) = (as[i..k⁻])
| . |      {append_segment}))
| . |    0
| . |
| . ⊢  A = (A[0..||A||⁻])
| .   by InvertRel 0
| .      THEN
| .      BackThru
| .      Thm* ∀as:T*. (as[0..||as||⁻]) = as    {whole_segment}
|
.\
. ⊢  ∀k:ℕ.
. |  (S:((A[0..||A||-j⁻]) @ (A[||A||-j..||A||-i⁻]↑k))
. |  @ (A[||A||-i..||A||⁻])←s)
. |  =
. |  (S:A←s)  by Analyze 6
. |
. 6. g : ℕN→S.car
. 7. InvFuns(S.car; ℕN; f; g)
. 8. A : Alph*
. 9. N<||A||
. 10. i : ℕ||A||
. 11. j : ℕ||A||
. 12. i<j
. 13. (λi.f((S:A[||A||-i..||A||⁻]←s)))(i)
. |   =
. |   (λi.f((S:A[||A||-i..||A||⁻]←s)))(j)
. ⊢   by Unfold ‘inv_funs‘ 7
. |
. 7. g o f = Id & f o g = Id
. 8. A : Alph*
. 9. N<||A||
. 10. i : ℕ||A||
. 11. j : ℕ||A||
. 12. i<j
. 13. (λi.f((S:A[||A||-i..||A||⁻]←s)))(i)
. |   =
. |   (λi.f((S:A[||A||-i..||A||⁻]←s)))(j)
. ⊢   by Analyze 7
. |
. 7. g o f = Id
. 8. f o g = Id
. 9. A : Alph*
. 10. N<||A||
```

```
. 11. i : ℕ||A||
. 12. j : ℕ||A||
. 13. i<j
. 14. (λi.f((S:A[||A||-i..||A||⁻]←s)))(i)
. |    =
. |    (λi.f((S:A[||A||-i..||A||⁻]←s)))(j)
. ⊢    by AbReduce 14
. |
. 14. f((S:A[||A||-i..||A||⁻]←s)) = f((S:A[||A||-j..||A||⁻]←s))
. ⊢    by ApplyFunToHypEquands g 14
. |
. 15. g(f((S:A[||A||-i..||A||⁻]←s))) = g(f((S:A[||A||-j..||A||⁻]←s)))
. ⊢    by RW (SweepDnC add_composeC) 15
. |
. 15. (g o f)((S:A[||A||-i..||A||⁻]←s))
. |    =
. |    (g o f)((S:A[||A||-j..||A||⁻]←s))
. ⊢    by RWH (SweepDnC (HypC 7)) 15
. |
. 15. Id((S:A[||A||-i..||A||⁻]←s)) = Id((S:A[||A||-j..||A||⁻]←s))
. ⊢    by AbReduce 15
. |
. 15. (S:A[||A||-i..||A||⁻]←s) = (S:A[||A||-j..||A||⁻]←s)
. ⊢    by Analyze 0
. |
. 16. k : ℕ
. ⊢   (S:((A[0..||A||-j⁻]) @ (A[||A||-j..||A||-i⁻]↑k))
. |   @ (A[||A||-i..||A||⁻])←s)
. |   =
. |   (S:A←s)  by NatInd 16
. |
. |\
. | .....basecase .....
. | 15. (S:A[||A||-i..||A||⁻]←s) = (S:A[||A||-j..||A||⁻]←s)
. | ⊢   (S:((A[0..||A||-j⁻]) @ (A[||A||-j..||A||-i⁻]↑0))
. | |   @ (A[||A||-i..||A||⁻])←s)
. | |   =
. | |   (S:A←s)  by RecCaseSplit ‘lpower‘
. | |
. | |\
. | | .....truecase .....
. | | 16. 0 = 0
. | | ⊢   (S:((A[0..||A||-j⁻]) @ nil) @ (A[||A||-i..||A||⁻])←s)
. | | |   =
. | | |   (S:A←s)
. | | | by RWH
. | | |    (LemmaC Thm* ∀as:T*. (as @ nil) = as    {append_back_nil})
. | | |    0
. | | |
. | | ⊢   (S:(A[0..||A||-j⁻]) @ (A[||A||-i..||A||⁻])←s) = (S:A←s)
. | | | by Inst
. | | |    Thm* ∀S:ActionSet(Alph), s:S.car, L1,L2,L:Alph*.
. | | |         (S:L1←s) = (S:L2←s) ⇒ (S:L @ L1←s) = (S:L @ L2←s)
. | | |    {maction_lemma}
. | | |    [Alph
```

```
. | | |    ;S
. | | |    ;s
. | | |    ;A[||A||-i..||A||⁻]
. | | |    ;A[||A||-j..||A||⁻]
. | | |    ;A[0..||A||-j⁻]]
. | | |
. | | |\
. | | | .....antecedent .....
. | | | ⊢  (S:A[||A||-i..||A||⁻]←s) = (S:A[||A||-j..||A||⁻]←s)
. | | |   by Auto
. | | |
. | | .\
. | | . 17. (S:(A[0..||A||-j⁻]) @ (A[||A||-i..||A||⁻])←s)
. | | . |   =
. | | . |   (S:(A[0..||A||-j⁻]) @ (A[||A||-j..||A||⁻])←s)
. | | . ⊢   by RWH (HypC 17) 0
. | | . |
. | | . ⊢  (S:(A[0..||A||-j⁻]) @ (A[||A||-j..||A||⁻])←s) = (S:A←s)
. | | . | by RWH
. | | . |    (LemmaC
. | | . |     Thm* ∀as:T*, i:{0...||as||}, j:{i...||as||}
. | | . |          , k:{j...||as||}.
. | | . |          ((as[i..j⁻]) @ (as[j..k⁻])) = (as[i..k⁻])
. | | . |     {append_segment})
. | | . |    0
. | | . |
. | | . ⊢  (S:A[0..||A||⁻]←s) = (S:A←s)
. | | .   by RWH
. | | .      (LemmaC
. | | .       Thm* ∀as:T*. (as[0..||as||⁻]) = as    {whole_segment})
. | | .      0
. | |
. | .\
. | . 16. ¬0 = 0
. | . ⊢  (S:((A[0..||A||-j⁻])
. | .        @ (A[||A||-j..||A||-i⁻]↑-1)
. | .        @ (A[||A||-j..||A||-i⁻]))
. | .    @ (A[||A||-i..||A||⁻])←s)
. | .    =
. | .    (S:A←s)  by Auto
. |
. .\
. . .....upcase .....
. . 16. k : ℤ
. . 17. 0<k
. . 18. (S:((A[0..||A||-j⁻]) @ (A[||A||-j..||A||-i⁻]↑k-1))
. . |   @ (A[||A||-i..||A||⁻])←s)
. . |   =
. . |   (S:A←s)
. . ⊢   by RecCaseSplit ‘lpower‘
. . |
. . |\
. . | .....truecase .....
. . | 19. k = 0
. . | ⊢  (S:((A[0..||A||-j⁻]) @ nil) @ (A[||A||-i..||A||⁻])←s)
```

```
. . |    =
. . |    (S:A←s)  by Auto
. . |
. . .\
. . . .....falsecase .....
. . . 19. ¬k = 0
. . . ⊢  (S:((A[0..||A||-j⁻])
. . . |      @ (A[||A||-j..||A||-i⁻]↑k-1)
. . . |      @ (A[||A||-j..||A||-i⁻]))
. . . |  @ (A[||A||-i..||A||⁻])←s)
. . . |  =
. . . |  (S:A←s)
. . . | by RW
. . . |    (LowerC
. . . |     (LemmaC
. . . |      Thm* ∀as,bs,cs:T*. ((as @ bs) @ cs) = (as @ (bs @ cs))
. . . |      {append_assoc}))
. . . |    0
. . . |
. . . ⊢  (S:(A[0..||A||-j⁻])
. . . |  @ ((A[||A||-j..||A||-i⁻]↑k-1) @ (A[||A||-j..||A||-i⁻]))
. . . |  @ (A[||A||-i..||A||⁻])←s)
. . . |  =
. . . |  (S:A←s)
. . . | by RWH
. . . |    (LemmaC
. . . |     Thm* ∀as,bs,cs:T*. ((as @ bs) @ cs) = (as @ (bs @ cs))
. . . |     {append_assoc})
. . . |    0
. . . |
. . . ⊢  (S:(A[0..||A||-j⁻])
. . . |  @ (A[||A||-j..||A||-i⁻]↑k-1)
. . . |  @ (A[||A||-j..||A||-i⁻])
. . . |  @ (A[||A||-i..||A||⁻])←s)
. . . |  =
. . . |  (S:A←s)
. . . | by RWH
. . . |    (LemmaC
. . . |     Thm* ∀as:T*, i:{0...||as||}, j:{i...||as||}
. . . |          , k:{j...||as||}.
. . . |          ((as[i..j⁻]) @ (as[j..k⁻])) = (as[i..k⁻])
. . . |     {append_segment})
. . . |    0
. . . |
. . . ⊢  (S:(A[0..||A||-j⁻])
. . . |  @ (A[||A||-j..||A||-i⁻]↑k-1)
. . . |  @ (A[||A||-j..||A||⁻])←s)
. . . |  =
. . . |  (S:A←s)
. . . | by RWH
. . . |    (RevLemmaC
. . . |     Thm* ∀as,bs,cs:T*. ((as @ bs) @ cs) = (as @ (bs @ cs))
. . . |     {append_assoc})
. . . |    0
. . . |
```

```
. . . ⊢  (S:((A[0..||A||-j⁻]) @ (A[||A||-j..||A||-i⁻]↑k-1))
. . .    @ (A[||A||-j..||A||⁻])←s)
. . .    =
. . .    (S:A←s)
. . .   by Inst
. . .      Thm* ∀S:ActionSet(Alph), s:S.car, L1,L2,L:Alph*.
. . .           (S:L1←s) = (S:L2←s) ⇒ (S:L @ L1←s) = (S:L @ L2←s)
. . .      {maction_lemma}
. . .      [Alph
. . .      ;S
. . .      ;s
. . .      ;A[||A||-i..||A||⁻]
. . .      ;A[||A||-j..||A||⁻]
. . .      ;(A[0..||A||-j⁻]) @ (A[||A||-j..||A||-i⁻]↑k-1)]
END
```