```
negnegelim_imp_notfin_imp_unb_inf  THM

⊢  (∀P:Prop. ¬¬P ⇒ P) ⇒ (∀A:Type. ¬Finite(A) ⇒ Unbounded(A))
| by Def of Finite(A) | Unbounded(A)
|
1. ∀P:Prop. ¬¬P ⇒ P
2. A : Type
3. ¬(∃n:ℕ. A ∼ ℕn)
4. k : ℕ
⊢  ∃∈(ℕk inj→ A)  by NatInd Hyp:-1
|
|\
| .....basecase .....
| 3. ¬(∃n:ℕ. A ∼ ℕn)
| ⊢  ∃∈(ℕ0 inj→ A)  by Witness: λx.x THEN Analyze THEN Analyze
|
.\
. .....upcase .....
. 4. k : ℤ
. 5. 0<k
. 6. ∃∈(ℕ(k-1) inj→ A)
. ⊢  ∃∈(ℕk inj→ A)
. | by New:f Analyze-1 THEN ∃a:A. ∀i:ℕ(k-1). f(i) ≠ a  Asserted
. |    THEN
. |    Analyze6
. |
. |\
. | 6. f : ℕ(k-1)→A
. | 7. Inj(ℕ(k-1); A; f)  [not for witness]
. | ⊢  ∃a:A. ∀i:ℕ(k-1). f(i) ≠ a  by BackThru: Hyp:1
. | |
. | ⊢  ¬¬(∃a:A. ∀i:ℕ(k-1). f(i) ≠ a)  by SimilarTo: Hyp:3
. | |
. | 8. ¬(∃a:A. ∀i:ℕ(k-1). f(i) ≠ a)
. | ⊢  ∃n:ℕ. A ∼ ℕn
. | | by FwdThru:
. | |    Thm*  ¬(∃x:T. Q(x)) ⇒ (∀x:T. ¬Q(x))    {not_over_exists_imp}
. | |    on [ Hyp:-1 ]
. | |    THEN
. | |    Witness: k-1
. | |
. | 8. ∀a:A. ¬(∀i:ℕ(k-1). f(i) ≠ a)
. | ⊢  A ∼ ℕ(k-1)
. | | by BackThru:
. | |    Thm*  (∃f:(A→B). Bij(A; B; f)) ⟺ (B ∼ A)
. | |    {bij_iff_1_1_corr_version2a}
. | |    THEN
. | |    Witness: f
. | |
. | ⊢  Bij(ℕ(k-1); A; f)  by Analyze
. | |
. | |\
. | | ⊢  Inj(ℕ(k-1); A; f)  by Unhide
. | |
. | .\
```

```
. | . ⊢  Surj(ℕ(k-1); A; f)  by SimilarTo: Hyp:8
. | . |
. | . 8. b : A
. | . 9. ¬(∀i:ℕ(k-1). f(i) ≠ b)
. | . ⊢  ∃a:ℕ(k-1). f(a) = b  by BackThru: Hyp:1 THEN SimilarTo Hyp:9
. | . |
. | . 9. ¬(∃a:ℕ(k-1). f(a) = b)
. | . ⊢  ∀i:ℕ(k-1). f(i) ≠ b
. | .   by BackThru:
. | .      Thm*  ¬(∃x:T. Q(x)) ⇒ (∀x:T. ¬Q(x))
. | .      {not_over_exists_imp}
. |
. .\
. . 6. f : ℕ(k-1)→A
. . 7. Inj(ℕ(k-1); A; f)  [not for witness]
. . 8. ∃a:A. ∀i:ℕ(k-1). f(i) ≠ a
. . ⊢  ∃∈(ℕk inj→ A)  by ExistHD Hyp:-1
. . |
. . 8. a : A
. . 9. ∀i:ℕ(k-1). f(i) ≠ a
. . ⊢  ∃∈(ℕk inj→ A)
. . | by Witness: λj.if j=_2k-1→ a else f(j) fi with type ℕk→A
. . |
. . |\
. . | .....assertion .....
. . | ⊢  (λj.if j=_2k-1→ a else f(j) fi) ∈ ℕk→A
. . |   by Analyze THEN SplitITE Concl
. . |
. . .\
. . . 7. Inj(ℕ(k-1); A; f)
. . . 8. a : A
. . . 9. ∀i:ℕ(k-1). f(i) ≠ a
. . . 10. (λj.if j=_2k-1→ a else f(j) fi) ∈ ℕk→A
. . . ⊢  (λj.if j=_2k-1→ a else f(j) fi) ∈ ℕk inj→ A  by Analyze
. . . |
. . . .....set predicate .....
. . . ⊢  Inj(ℕk; A; λj.if j=_2k-1→ a else f(j) fi)
. . . | by UnivCD THEN Reduce Hyp:-1
. . . |
. . . 11. a1 : ℕk
. . . 12. a2 : ℕk
. . . 13. if a1=_2k-1→ a else f(a1) fi = if a2=_2k-1→ a else f(a2) fi
. . . ⊢  a1 = a2  by SplitITE Hyp:-1 THEN SplitITE Hyp:-2
. . . |
. . . |\
. . . | .....falsecase .....
. . . | 13. a = f(a2)

. . . | 14. a1 = k-1
. . . | 15. a2 ≠ k-1
. . . | ⊢  a1 = a2
. . . |   by False  Asserted THEN BackThru: Hyp:9 Using:[a2]
. . . |
. . . |\
. . . | .....truecase .....
. . . | 13. f(a1) = a
```

```
. . . | 14. a1 ≠ k-1
. . . | 15. a2 = k-1
. . . | ⊢  a1 = a2
. . . |   by False  Asserted THEN BackThru: Hyp:9 Using:[a1]
. . . |
. . . .\
. . . . .....falsecase .....
. . . . 13. f(a1) = f(a2)
. . . . 14. a1 ≠ k-1
. . . . 15. a2 ≠ k-1
. . . . ⊢  a1 = a2  by BackThru: Inj(ℕ(k-1); A; f)
END
```