```
trivial-arrows  THM

⊢  ∀k:ℕ+ , L:ℕ List. k-1->L^k ⟺ (∃i:ℕ||L||. L[i]<k)
| by Auto THEN (All (Unfold `arrows`)) THEN ExRepD
|
|\
| 1. k : ℕ+
| 2. L : ℕ List
| 3. ∀n:ℕ.
| |  k-1≤n
| |  ⇒
| |  (∀G:({s:(ℕn List)| ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }
| |       →ℕ||L||).
| |
| |   ∃c:ℕ||L||, f:(ℕL[c]→ℕn).
| |   increasing(f;L[c])
| |   & (∀s:ℕL[c] List.
| |      ||s|| = k
| |      ⇒
| |      (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) ⇒ G(map(f;s)) = c))
| ⊢  ∃i:ℕ||L||. L[i]<k
| | by (InstHyp [k-1;λs.•] -1) THEN (Try (Complete Auto))
| |
| |\
| | .....wf .....
| | 1. k : ℕ+
| | 2. L : ℕ List
| | 3. ∀n:ℕ.
| | |  k-1≤n
| | |  ⇒
| | |  (∀G:({s:(ℕn List)
| | |        | ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }
| | |       →ℕ||L||).
| | |
| | |   ∃c:ℕ||L||, f:(ℕL[c]→ℕn).
| | |   increasing(f;L[c])
| | |   & (∀s:ℕL[c] List.
| | |      ||s|| = k
| | |      ⇒
| | |      (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) ⇒ G(map(f;s)) = c))
| | ⊢  (λs.•)
| | |  ∈ {s:(ℕ(k-1) List)
| | |     | ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }
| | |    →ℕ||L||  by Analyze THEN (Try (Complete Auto))
| | |
| | .....subterm .....
| | 1. k : ℕ+
| | 2. L : ℕ List
| | 3. ∀n:ℕ.
| | |  k-1≤n
| | |  ⇒
| | |  (∀G:({s:(ℕn List)
| | |        | ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }
| | |       →ℕ||L||).
| | |
```

```
| | |   ∃c:ℕ||L||, f:(ℕL[c]→ℕn).
| | |   increasing(f;L[c])
| | |   & (∀s:ℕL[c] List.
| | |      ||s|| = k
| | |      ⇒
| | |      (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) ⇒ G(map(f;s)) = c))
| | 4. {s:(ℕ(k-1) List)| ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }
| | ⊢  · ∈ ℕ||L||  by (Analyze -1) THEN (Analyze -1)
| | |
| | 1. k : ℕ+
| | 2. L : ℕ List
| | 3. ∀n:ℕ.
| | |  k-1≤n
| | |  ⇒
| | |  (∀G:({s:(ℕn List)
| | |        | ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }
| | |       →ℕ||L||).
| | |
| | |   ∃c:ℕ||L||, f:(ℕL[c]→ℕn).
| | |   increasing(f;L[c])
| | |   & (∀s:ℕL[c] List.
| | |      ||s|| = k
| | |      ⇒
| | |      (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) ⇒ G(map(f;s)) = c))
| | 4. s : ℕ(k-1) List
| | 5. ||s|| = k
| | 6. ∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]
| | ⊢  · ∈ ℕ||L||
| | | by Inst
| | |    Thm* ∀n,m:ℕ, f:(ℕn→ℕm). Inj(ℕn; ℕm; f) ⇒ n≤m
| | |    {pigeon_DASH_hole}
| | |    [k;k-1;λx.s[x]]
| | |
| | |\
| | | .....antecedent .....
| | | 1. k : ℕ+
| | | 2. L : ℕ List
| | | 3. ∀n:ℕ.
| | |    k-1≤n
| | |    ⇒
| | |    (∀G:({s:(ℕn List)
| | |          | ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }
| | |         →ℕ||L||).
| | |
| | |     ∃c:ℕ||L||, f:(ℕL[c]→ℕn).
| | |     increasing(f;L[c])
| | |     & (∀s:ℕL[c] List.
| | |        ||s|| = k
| | |        ⇒
| | |        (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) ⇒ G(map(f;s)) = c))
| | | 4. s : ℕ(k-1) List
| | | 5. ||s|| = k
| | | 6. ∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]
| | | ⊢  Inj(ℕk; ℕ(k-1); λx.s[x])
| | |   by (BackThru
```

```
| | |       Thm* ∀k,m:ℕ, f:(ℕk→ℕm). increasing(f;k) ⇒ Inj(ℕk; ℕm; f)
| | |       {increasing_inj})
| | |      THEN
| | |      (Analyze 0)
| | |      THEN
| | |      (Reduce 0)
| | |      THEN
| | |      EasyHyp
| | |
| | .\
| | . .....wf .....
| | . 1. k : ℕ+
| | . 2. L : ℕ List
| | . 3. ∀n:ℕ.
| | .    k-1≤n
| | .    ⇒
| | .    (∀G:({s:(ℕn List)
| | .          | ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }
| | .         →ℕ||L||).
| | .
| | .     ∃c:ℕ||L||, f:(ℕL[c]→ℕn).
| | .     increasing(f;L[c])
| | .     & (∀s:ℕL[c] List.
| | .        ||s|| = k
| | .        ⇒
| | .        (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) ⇒ G(map(f;s)) = c))
| | . 4. s : ℕ(k-1) List
| | . 5. ||s|| = k
| | . 6. ∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]
| | . 7. k≤k-1
| | . ⊢ · ∈ ℤ  by ObviousContradiction [-1]
| |
| .\
| . 1. k : ℕ+
| . 2. L : ℕ List
| . 3. ∀n:ℕ.
| . |  k-1≤n
| . |  ⇒
| . |  (∀G:({s:(ℕn List)
| . |        | ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }
| . |       →ℕ||L||).
| . |
| . |   ∃c:ℕ||L||, f:(ℕL[c]→ℕn).
| . |   increasing(f;L[c])
| . |   & (∀s:ℕL[c] List.
| . |      ||s|| = k
| . |      ⇒
| . |      (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) ⇒ G(map(f;s)) = c))
| . 4. ∃c:ℕ||L||, f:(ℕL[c]→ℕ(k-1)).
| . |  increasing(f;L[c])
| . |  & (∀s:ℕL[c] List.
| . |     ||s|| = k
| . |     ⇒
| . |     (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) ⇒ (λs.·)(map(f;s)) = c)
| . ⊢  ∃i:ℕ||L||. L[i]<k  by (ParallelOp -1) THEN ExRepD
```

```
| . |
| . 1. k : ℕ+
| . 2. L : ℕ List
| . 3. ∀n:ℕ.
| .    k-1≤n
| .    ⇒
| .    (∀G:({s:(ℕn List)
| .          | ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }
| .         →ℕ||L||).
| .
| .     ∃c:ℕ||L||, f:(ℕL[c]→ℕn).
| .     increasing(f;L[c])
| .     & (∀s:ℕL[c] List.
| .        ||s|| = k
| .        ⇒
| .        (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) ⇒ G(map(f;s)) = c))
| . 4. c : ℕ||L||
| . 5. f : ℕL[c]→ℕ(k-1)
| . 6. increasing(f;L[c])
| . 7. ∀s:ℕL[c] List.
| .    ||s|| = k
| .    ⇒
| .    (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) ⇒ (λs.•)(map(f;s)) = c
| . ⊢  L[c]<k
| .   by (Inst
| .       Thm* ∀n,m:ℕ, f:(ℕn→ℕm). Inj(ℕn; ℕm; f) ⇒ n≤m
| .       {pigeon_DASH_hole}
| .       [L[c];k-1;f])
| .      THEN
| .      (BackThru
| .       Thm* ∀k,m:ℕ, f:(ℕk→ℕm). increasing(f;k) ⇒ Inj(ℕk; ℕm; f)
| .       {increasing_inj})
|
.\
. 1. k : ℕ+
. 2. L : ℕ List
. 3. i : ℕ||L||
. 4. L[i]<k
. 5. n : ℕ
. 6. k-1≤n
. 7. G :
. |  {s:(ℕn List)| ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }→
. |  ℕ||L||
. ⊢  ∃c:ℕ||L||, f:(ℕL[c]→ℕn).
. |  increasing(f;L[c])
. |  & (∀s:ℕL[c] List.
. |     ||s|| = k
. |     ⇒
. |     (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) ⇒ G(map(f;s)) = c)
. | by ((InstConcl [i;λx.x]) THEN (Reduce 0))
. |    THENA
. |    (Try (Complete Auto))
. |
. |\
. | 1. k : ℕ+
```

```
. | 2. L : ℕ List
. | 3. i : ℕ||L||
. | 4. L[i]<k
. | 5. n : ℕ
. | 6. k-1≤n
. | 7. G :
. | |  {s:(ℕn List)| ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }⟶
. | |  ℕ||L||
. | ⊢  increasing(λx.x;L[i])
. | |  & (∀s:ℕL[i] List.
. | |     ||s|| = k
. | |     ⇒
. | |     (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) ⇒ G(map(λx.x;s)) = i)
. | | by Analyze 0
. | |
. | |\
. | | 1. k : ℕ+
. | | 2. L : ℕ List
. | | 3. i : ℕ||L||
. | | 4. L[i]<k
. | | 5. n : ℕ
. | | 6. k-1≤n
. | | 7. {s:(ℕn List)| ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }⟶
. | |    ℕ||L||
. | | ⊢  increasing(λx.x;L[i])
. | |   by BackThru Thm* ∀k:ℕ. increasing(λi.i;k)    {id_increasing}
. | |
. | .\
. | . 1. k : ℕ+
. | . 2. L : ℕ List
. | . 3. i : ℕ||L||
. | . 4. L[i]<k
. | . 5. n : ℕ
. | . 6. k-1≤n
. | . 7. G :
. | . |  {s:(ℕn List)| ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }⟶
. | . |  ℕ||L||
. | . 8. increasing(λx.x;L[i])
. | . ⊢  ∀s:ℕL[i] List.
. | . |  ||s|| = k
. | . |  ⇒
. | . |  (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) ⇒ G(map(λx.x;s)) = i
. | . | by Auto
. | . |    THEN
. | . |    (Inst
. | . |     Thm* ∀n,m:ℕ, f:(ℕn→ℕm). Inj(ℕn; ℕm; f) ⇒ n≤m
. | . |     {pigeon_DASH_hole}
. | . |     [k;L[i];λx.s[x]])
. | . |
. | . .....antecedent .....
. | . 1. k : ℕ+
. | . 2. L : ℕ List
. | . 3. i : ℕ||L||
. | . 4. L[i]<k
. | . 5. n : ℕ
```

```
. | . 6. k-1≤n
. | . 7. {s:(ℕn List)| ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }→
. | .  |  ℕ||L||
. | . 8. increasing(λx.x;L[i])
. | . 9. s : ℕL[i] List
. | . 10. ||s|| = k
. | . 11. ∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]
. | . ⊢  Inj(ℕk; ℕL[i]; λx.s[x])
. | . | by BackThru
. | . |    Thm* ∀k,m:ℕ, f:(ℕk→ℕm). increasing(f;k) ⇒ Inj(ℕk; ℕm; f)
. | . |    {increasing_inj}
. | . |
. | . 1. k : ℕ+
. | . 2. L : ℕ List
. | . 3. i : ℕ||L||
. | . 4. L[i]<k
. | . 5. n : ℕ
. | . 6. k-1≤n
. | . 7. {s:(ℕn List)| ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }→
. | .    ℕ||L||
. | . 8. increasing(λx.x;L[i])
. | . 9. s : ℕL[i] List
. | . 10. ||s|| = k
. | . 11. ∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]
. | . ⊢  increasing(λx.s[x];k)
. | .   by (Analyze 0) THEN (Reduce 0) THEN EasyHyp
. |

. |\
. | .....wf .....
. | 1. k : ℕ+
. | 2. L : ℕ List
. | 3. i : ℕ||L||
. | 4. L[i]<k
. | 5. n : ℕ
. | 6. k-1≤n
. | 7. G :
. |    {s:(ℕn List)| ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }→
. |    ℕ||L||
. | 8. f : ℕL[i]→ℕn
. | ⊢  (increasing(f;L[i])
. |     & (∀s:ℕL[i] List.
. |        ||s|| = k
. |        ⇒
. |        (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) ⇒ G(map(f;s)) = i))
. |    ∈ Prop
. |   by ((Auto THEN Analyze THEN (Try (Complete Auto))
. |        THEN
. |        (RWO
. |         Thm* ∀f:(A→B), as:A List. ||map(f;as)|| = ||as|| ∈ ℤ
. |         {map_length}
. |         0))
. |       THEN
. |       (RWO
. |        Thm* ∀f:(A→B), as:A List, n:ℕ||as||.
. |             map(f;as)[n] = f(as[n])
```

```
. |        {map_select}
. |        0))
. |      THEN
. |      (AllHyps
. |       (λh.
. |        FwdThru
. |        Thm* ∀k:ℕ, f:(ℕk→ℤ).
. |             increasing(f;k) ⇒ (∀x,y:ℕk. x<y ⇒ f(x)<f(y))
. |        {increasing_implies}
. |        [h]))
. |      THEN
. |      (BackThru -1)
. |      THEN
. |      EasyHyp
. |
. .\
. . .....wf .....
. . 1. k : ℕ+
. . 2. L : ℕ List
. . 3. i : ℕ||L||
. . 4. L[i]<k
. . 5. n : ℕ
. . 6. k-1≤n
. . 7. G :
. .    {s:(ℕn List)| ||s|| = k & (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) }→
. .    ℕ||L||
. . 8. c : ℕ||L||
. . ⊢  (∃f:(ℕL[c]→ℕn).
. .     increasing(f;L[c])
. .     & (∀s:ℕL[c] List.
. .        ||s|| = k
. .        ⇒
. .        (∀x,y:ℕ||s||. x<y ⇒ s[x]<s[y]) ⇒ G(map(f;s)) = c))
. .    ∈ Prop
. .   by ((Auto THEN Analyze THEN (Try (Complete Auto))
. .        THEN
. .        (RWO
. .         Thm* ∀f:(A→B), as:A List. ||map(f;as)|| = ||as|| ∈ ℤ
. .         {map_length}
. .         0))
. .       THEN
. .       (RWO
. .        Thm* ∀f:(A→B), as:A List, n:ℕ||as||.
. .             map(f;as)[n] = f(as[n])
. .        {map_select}
. .        0))
. .      THEN
. .      (AllHyps
. .       (λh.
. .        FwdThru
. .        Thm* ∀k:ℕ, f:(ℕk→ℤ).
. .             increasing(f;k) ⇒ (∀x,y:ℕk. x<y ⇒ f(x)<f(y))
. .        {increasing_implies}
. .        [h]))
. .      THEN
```

```
. .       (BackThru -1)
. .       THEN
. .       EasyHyp
END
```