```
select_upto  THM

⊢  ∀i,j:ℕ, k:ℕ(j-i). upto(i;j)[k] = i+k
| by Assert (∀d,i,j:ℕ, k:ℕd. j-i = d ⇒ upto(i;j)[k] = i+k)
|
|\
| .....assertion .....
| ⊢  ∀d,i,j:ℕ, k:ℕd. j-i = d ⇒ upto(i;j)[k] = i+k
| | by InductionOnNat THEN (RecUnfold `upto` 0) THEN SplitOnConclITE
| |    THEN
| |    (CaseNat 0 `k`)
| |    THEN
| |    (Reduce 0)
| |
| 1. d : ℤ
| 2. 0<d
| 3. ∀i,j:ℕ, k:ℕ(d-1). j-i = d-1 ⇒ upto(i;j)[k] = i+k
| 4. i : ℕ
| 5. j : ℕ
| 6. k : ℕd
| 7. j-i = d
| 8. i<j
| 9. ¬k = 0
| ⊢  [i / upto(i+1;j)][k] = i+k
| | by RWO
| |    Thm* ∀a:T, as:T List, i:ℤ.
| |         0<i ⇒ i≤||as|| ⇒ [a / as][i] = as[(i-1)]
| |    {select_cons_tl}
| |    0
| |
| |\
| | .....rewrite subgoal .....
| | 1. d : ℤ
| | 2. 0<d
| | 3. ∀i,j:ℕ, k:ℕ(d-1). j-i = d-1 ⇒ upto(i;j)[k] = i+k
| | 4. i : ℕ
| | 5. j : ℕ
| | 6. k : ℕd
| | 7. j-i = d
| | 8. i<j
| | 9. ¬k = 0
| | ⊢  k≤||upto(i+1;j)||
| |   by RWO Thm* ∀j:ℕ, i:ℕj. ||upto(i;j)|| = j-i ∈ ℕ    {length_upto}
| |      0
| |
| .\
| . 1. d : ℤ
| . 2. 0<d
| . 3. ∀i,j:ℕ, k:ℕ(d-1). j-i = d-1 ⇒ upto(i;j)[k] = i+k
| . 4. i : ℕ
| . 5. j : ℕ
| . 6. k : ℕd
| . 7. j-i = d
| . 8. i<j
| . 9. ¬k = 0
```

```
| . ⊢  upto(i+1;j)[(k-1)] = i+k  by InstHyp [i+1;j;k-1] 3
|
.\
. 1. ∀d,i,j:ℕ, k:ℕd. j-i = d ⇒ upto(i;j)[k] = i+k
. ⊢  ∀i,j:ℕ, k:ℕ(j-i). upto(i;j)[k] = i+k
.   by Auto THEN (InstHyp [j-i;i;j;k] 1)
END
```