```
mklist_select  THM

⊢  ∀T:Type, n:ℕ, f:(ℕn→T), i:ℕn. mklist(n;f)[i] = f(i)
| by RepeatFor 2 (D 0 THENA Auto) THEN NatInd (-1) THEN Reduce 0
|
|\
| .....basecase .....
| 1. T : Type
| ⊢  ∀f:(ℕ0→T), i:ℕ0. mklist(0;f)[i] = f(i)  by Auto
|
.\
. .....upcase .....
. 1. T : Type
. 2. n : ℤ
. 3. 0<n
. 4. ∀f:(ℕ(n-1)→T), i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. ⊢  ∀f:(ℕn→T), i:ℕn. mklist(n;f)[i] = f(i)  by ParallelOp (-1)
. |
. 1. T : Type
. 2. n : ℤ
. 3. 0<n
. 4. ∀f:(ℕ(n-1)→T), i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. 5. f : ℕn→T
. 6. ∀i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. ⊢  ∀i:ℕn. mklist(n;f)[i] = f(i)
. | by ...
. THENA Auto
. |
. |\
. | .....truecase .....
. | 1. T : Type
. | 2. n : ℤ
. | 3. 0<n
. | 4. ∀f:(ℕ(n-1)→T), i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. | 5. f : ℕn→T
. | 6. ∀i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. | 7. n = 0
. | ⊢  ∀i:ℕn. nil[i] = f(i)  by Auto
. |
. .\
. . .....falsecase .....
. . 1. T : Type
. . 2. n : ℤ
. . 3. 0<n
. . 4. ∀f:(ℕ(n-1)→T), i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. . 5. f : ℕn→T
. . 6. ∀i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. . 7. ¬n = 0
. . ⊢  ∀i:ℕn.
. . |  ((λi,l. l @ [(f(i))])
. . |   (n-1
. . |   ,primrec(n-1;nil;λi,l. l @ [(f(i))])))[i]
. . |  =
. . |  f(i)  by Reduce 0 THEN Fold `mklist` 0 THEN Auto
. . |
```

```
. . 1. T : Type
. . 2. n : ℤ
. . 3. 0<n
. . 4. ∀f:(ℕ(n-1)→T), i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. . 5. f : ℕn→T
. . 6. ∀i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. . 7. ¬n = 0
. . 8. i : ℕn
. . ⊢  (mklist(n-1;f) @ [(f(n-1))])[i] = f(i)
. . | by Decide i<n-1 THENA Auto
. . |
. . |\
. . | 1. T : Type
. . | 2. n : ℤ
. . | 3. 0<n
. . | 4. ∀f:(ℕ(n-1)→T), i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. . | 5. f : ℕn→T
. . | 6. ∀i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. . | 7. ¬n = 0
. . | 8. i : ℕn
. . | 9. i<n-1
. . | ⊢  (mklist(n-1;f) @ [(f(n-1))])[i] = f(i)
. . | | by RWO "select_append_front" 0 THEN Auto
. . | |
. . | |\
. . | | 1. T : Type
. . | | 2. n : ℤ
. . | | 3. 0<n
. . | | 4. ∀f:(ℕ(n-1)→T), i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. . | | 5. f : ℕn→T
. . | | 6. ∀i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. . | | 7. ¬n = 0
. . | | 8. i : ℕn
. . | | 9. i<n-1
. . | | ⊢  i<||mklist(n-1;f)||  by RWO "mklist_length" 0 THEN Auto
. . | |
. . | .\
. . | . 1. T : Type
. . | . 2. n : ℤ
. . | . 3. 0<n
. . | . 4. ∀f:(ℕ(n-1)→T), i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. . | . 5. f : ℕn→T
. . | . 6. ∀i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. . | . 7. ¬n = 0
. . | . 8. i : ℕn
. . | . 9. i<n-1
. . | . ⊢  mklist(n-1;f)[i] = f(i)  by EasyHyp

. . |
. . .\
. . . 1. T : Type
. . . 2. n : ℤ
. . . 3. 0<n
. . . 4. ∀f:(ℕ(n-1)→T), i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. . . 5. f : ℕn→T
. . . 6. ∀i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
```

```
. . . 7. ¬n = 0
. . . 8. i : ℕn
. . . 9. ¬i<n-1
. . . ⊢  (mklist(n-1;f) @ [(f(n-1))])[i] = f(i)
. . . | by Subst' i = n-1 0 THENA Auto'
. . .
. . . |    THEN RWO "select_append_back" 0 THEN Auto
. . . THEN Subst'
. . . |    ||mklist(n-1;f)|| = n-1 0 THEN Auto'
. . .
. . . |    THEN Try (BackThruLemma `mklist_length` THEN Auto)
. . . |
. . . 1. T : Type
. . . 2. n : ℤ
. . . 3. 0<n
. . . 4. ∀f:(ℕ(n-1)→T), i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. . . 5. f : ℕn→T
. . . 6. ∀i:ℕ(n-1). mklist(n-1;f)[i] = f(i)
. . . 7. ¬n = 0
. . . 8. i : ℕn
. . . 9. ¬i<n-1
. . . ⊢  [(f(n-1))][(n-1-(n-1))] = f(n-1)
. . .   by Subst' n-1-(n-1) = 0 0 THEN Reduce 0 THEN Auto'
END
```