```
primrec_list_accum  THM

⊢  ∀n:ℕ, f,x:Top. primrec(n;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n))
| by (Analyze 0) THEN (NatInd -1)
|
|\
| .....basecase .....
| ⊢  ∀f,x:Top. primrec(0;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;0))
|   by (RecUnfold ‘upto‘ 0) THEN (Reduce 0)
|
.\
. .....upcase .....
. 1. n : ℤ
. 2. 0<n
. 3. ∀f,x:Top. primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. ⊢  ∀f,x:Top. primrec(n;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n))
. | by ((RepeatFor 2 (ParallelOp -1)) THEN (RecUnfold ‘primrec‘ 0)
. |     THEN
. |     SplitOnConclITE)
. |    THENL
. |    [Auto;Id]
. |
. .....falsecase .....
. 1. n : ℤ
. 2. 0<n
. 3. ∀f,x:Top. primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. 4. f : Top
. 5. ∀x:Top. primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. 6. x : Top
. 7. primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. 8. ¬n = 0
. ⊢  (f(n-1,primrec(n-1;x;f))) ~ list_accum(i,y.f(y,i);x;upto(0;n))
. | by HypSubst -2 0
. |
. 1. n : ℤ
. 2. 0<n
. 3. ∀f,x:Top. primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. 4. f : Top
. 5. ∀x:Top. primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. 6. x : Top
. 7. primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. 8. ¬n = 0
. ⊢  (f
. |   (n-1
. |   ,list_accum(i,y.f(y,i);x;upto(0;n
. |    -1)))) ~ list_accum(i,y.f(y,i);x;upto(0;n))
. | by Assert
. |    (∀i:ℕ.
. |     i<n
. |     ⇒
. |     (∀z:Top.
. |      (f
. |       (n-1
. |       ,list_accum(i,y.f(y,i);z;upto(n-1-i;n
. |        -1)))) ~ list_accum(i,y.f(y,i);z;upto(n-1-i;n))))
```

```
. |
. |\
. | .....assertion .....
. | 1. n : ℤ
. | 2. 0<n
. | 3. ∀f,x:Top.
. | |  primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | 4. f : Top
. | 5. ∀x:Top. primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | 6. x : Top
. | 7. primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | 8. ¬n = 0
. | ⊢  ∀i:ℕ.
. | |  i<n
. | |  ⇒
. | |  (∀z:Top.
. | |   (f
. | |    (n-1
. | |    ,list_accum(i,y.f(y,i);z;upto(n-1-i;n
. | |     -1)))) ~ list_accum(i,y.f(y,i);z;upto(n-1-i;n)))
. | | by (Analyze 0) THEN (NatInd -1)
. | |
. | |\
. | | .....basecase .....
. | | 1. n : ℤ
. | | 2. 0<n
. | | 3. ∀f,x:Top.
. | | |  primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | | 4. f : Top
. | | 5. ∀x:Top.
. | | |  primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | | 6. x : Top
. | | 7. primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | | 8. ¬n = 0
. | | ⊢  0<n
. | | |  ⇒
. | | |  (∀z:Top.
. | | |   (f
. | | |    (n-1
. | | |    ,list_accum(i,y.f(y,i);z;upto(n-1-0;n
. | | |     -1)))) ~ list_accum(i,y.f(y,i);z;upto(n-1-0;n)))
. | | | by (Analyze 0) THEN (Subst (upto(n-1-0;n-1) ~ nil) 0)
. | | |    THEN
. | | |    (Reduce 0)
. | | |
. | | |\
. | | | .....equality .....
. | | | 1. n : ℤ
. | | | 2. 0<n
. | | | 3. ∀f,x:Top.
. | | |    primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | | | 4. f : Top
. | | | 5. ∀x:Top.
. | | |    primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | | | 6. x : Top
```

```
. | | | 7. primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | | | 8. ¬n = 0
. | | | 9. 0<n
. | | | ⊢  upto(n-1-0;n-1) ~ nil
. | | |   by (RecUnfold ‘upto‘ 0) THEN SplitOnConclITE
. | | |
. | | .\
. | | . 1. n : ℤ
. | | . 2. 0<n
. | | . 3. ∀f,x:Top.
. | | . |  primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | | . 4. f : Top
. | | . 5. ∀x:Top.
. | | . |  primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | | . 6. x : Top
. | | . 7. primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | | . 8. ¬n = 0
. | | . 9. 0<n
. | | . ⊢  ∀z@0:Top.
. | | . |  (f(n-1,z@0)) ~ list_accum(i,y.f(y,i);z@0;upto(n-1-0;n))
. | | . | by (Subst (upto(n-1-0;n) ~ [(n-1)]) 0) THEN (Reduce 0)
. | | . |
. | | . |\
. | | . | .....equality .....
. | | . | 1. n : ℤ
. | | . | 2. 0<n
. | | . | 3. ∀f,x:Top.
. | | . |    primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | | . | 4. f : Top
. | | . | 5. ∀x:Top.
. | | . |    primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | | . | 6. x : Top
. | | . | 7. primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | | . | 8. ¬n = 0
. | | . | 9. 0<n
. | | . | ⊢  upto(n-1-0;n) ~ [(n-1)]
. | | . |   by (RecUnfold ‘upto‘ 0) THEN SplitOnConclITE
. | | . |      THEN
. | | . |      (Try (Complete Auto))
. | | . |      THEN
. | | . |      Analyze
. | | . |      THEN
. | | . |      (Try (Complete Auto))
. | | . |      THEN
. | | . |      (RecUnfold ‘upto‘ 0)
. | | . |      THEN
. | | . |      SplitOnConclITE
. | | . |      THEN
. | | . |      (Try (Complete Auto))
. | | . |
. | | . .\
. | | . . 1. n : ℤ
. | | . . 2. 0<n
. | | . . 3. ∀f,x:Top.
. | | . .    primrec(n-1;x;f) ~ list_accum(i,y.f(y,i);x;upto(0;n-1))
```

```
. | | . . 4. f : Top
. | | . . 5. ∀x:Top.
. | | . .    primrec(n-1;x;f) ∼ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | | . . 6. x : Top
. | | . . 7. primrec(n-1;x;f) ∼ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | | . . 8. ¬n = 0
. | | . . 9. 0<n
. | | . . ⊢  ∀z@0:Top. (f(n-1,z@0)) ∼ (f(n-1,z@0))  by Auto
. | |
. | .\
. | . .....upcase .....
. | . 1. n : ℤ
. | . 2. 0<n
. | . 3. ∀f,x:Top.
. | . |  primrec(n-1;x;f) ∼ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | . 4. f : Top
. | . 5. ∀x:Top.
. | . |  primrec(n-1;x;f) ∼ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | . 6. x : Top
. | . 7. primrec(n-1;x;f) ∼ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | . 8. ¬n = 0
. | . 9. i : ℤ
. | . 10. 0<i
. | . 11. i-1<n
. | . |   ⇒
. | . |   (∀z:Top.
. | . |    (f
. | . |     (n-1
. | . |     ,list_accum(i,y.f(y,i);z;upto(n-1-(i-1);n
. | . |      -1)))) ∼ list_accum(i,y.f(y,i);z;upto(n-1-(i-1);n)))
. | . ⊢  i<n
. | . |  ⇒
. | . |  (∀z:Top.
. | . |   (f
. | . |    (n-1
. | . |    ,list_accum(i,y.f(y,i);z;upto(n-1-i;n
. | . |     -1)))) ∼ list_accum(i,y.f(y,i);z;upto(n-1-i;n)))
. | . | by (ParallelOp -1) THEN (RecUnfold ‘upto‘ 0)
. | . |    THEN
. | . |    (RepeatFor 2
. | . |     (SplitOnConclITE THEN (Try (Complete Auto))
. | . |      THEN
. | . |      (Reduce 0)))
. | . |
. | . 1. n : ℤ
. | . 2. 0<n
. | . 3. ∀f,x:Top.
. | .    primrec(n-1;x;f) ∼ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | . 4. f : Top
. | . 5. ∀x:Top.
. | .    primrec(n-1;x;f) ∼ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | . 6. x : Top
. | . 7. primrec(n-1;x;f) ∼ list_accum(i,y.f(y,i);x;upto(0;n-1))
. | . 8. ¬n = 0
. | . 9. i : ℤ
```

```
. | . 10. 0<i
. | . 11. i<n
. | . 12. ∀z:Top.
. | .     (f
. | .      (n-1
. | .      ,list_accum(i,y.f(y,i);z;upto(n-1-(i-1);n
. | .       -1)))) ∼ list_accum(i,y.f(y,i);z;upto(n-1-(i-1);n))
. | . 13. n-1-i<n-1
. | . 14. n-1-i<n
. | . ⊢  ∀z:Top.
. | .    (f
. | .     (n-1
. | .     ,list_accum(i,y.f(y,i);f(n-1-i,z);upto(n-1-i+1;n
. | .      -1)))) ∼ list_accum(i,y.f(y,i);f(n-1-i,z);upto(n-1-i+1;n))
. | .   by (Analyze 0) THEN (InstHyp [f(n-1-i,z)] -4)
. | .      THEN
. | .      (NthHypSq -1)
. | .      THEN
. | .      (RepeatFor 4 Analyze)
. |
. .\
. . 1. n : ℤ
. . 2. 0<n
. . 3. ∀f,x:Top.
. .    primrec(n-1;x;f) ∼ list_accum(i,y.f(y,i);x;upto(0;n-1))
. . 4. f : Top
. . 5. ∀x:Top. primrec(n-1;x;f) ∼ list_accum(i,y.f(y,i);x;upto(0;n-1))
. . 6. x : Top
. . 7. primrec(n-1;x;f) ∼ list_accum(i,y.f(y,i);x;upto(0;n-1))
. . 8. ¬n = 0
. . 9. ∀i:ℕ.
. .    i<n
. .    ⇒
. .    (∀z:Top.
. .     (f
. .      (n-1
. .      ,list_accum(i,y.f(y,i);z;upto(n-1-i;n
. .       -1)))) ∼ list_accum(i,y.f(y,i);z;upto(n-1-i;n)))
. . ⊢  (f
. .     (n-1
. .     ,list_accum(i,y.f(y,i);x;upto(0;n
. .      -1)))) ∼ list_accum(i,y.f(y,i);x;upto(0;n))
. .   by (InstHyp [n-1;x] -1) THEN (NthHypSq -1)
. .      THEN
. .      (RepeatFor 4 Analyze)
END
```