```
accumulate_wf  THM

⊢  ∀A,B:Type, P:(B→A→𝔹), F,G,H:(B→A→A), N:(B→A→(B List))
|  , M:(A→ℕ).
|  (∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s))
|  ⇒
|  (∀i:B, s:A. M(G(i,s))≤M(s))
|  ⇒
|  (∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s))
|  ⇒
|  (∀j:B, u:A.
|   process u j  where
|   process s i ==
|   if P(i,s) then F(i,s)
|   else G(i,s) where
|        xs := N(i,s);
|        s:= H(i,s);
|        while not null xs {
|        s := process s (hd xs);
|        xs := tl xs;
|        }
|   ∈ {s:A| M(s)≤M(u) })
| by Auto THEN (RepeatFor 2 (MoveToConcl -1))
|    THEN
|    (Assert
|     (∀m:ℕ, u:A.
|      M(u) = m
|      ⇒
|      (∀j:B.
|       process u j  where
|       process s i ==
|       if P(i,s) then F(i,s)
|       else G(i,s) where
|            xs := N(i,s);
|            s:= H(i,s);
|            while not null xs {
|            s := process s (hd xs);
|            xs := tl xs;
|            }
|       ∈ {s:A| M(s)≤M(u) })))
|    THEN
|    (Try (Complete (Auto THEN (Using [‘m’,M(u)] EasyHyp))))
|    THEN
|    CompleteInductionOnNat
|    THEN
|    (RecUnfold ‘accumulate‘ 0)
|    THEN
|    SplitOnConclITE
|    THEN
|    (Try (Complete (Auto THEN Analyze THEN EasyHyp)))
|
.....falsecase .....
1. A : Type
2. B : Type
3. P : B→A→𝔹
```

```
4. F : B→A→A
5. G : B→A→A
6. H : B→A→A
7. N : B→A→(B List)
8. M : A→ℕ
9. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
10. ∀i:B, s:A. M(G(i,s))≤M(s)
11. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
12. m : ℕ
13. ∀m1:ℕ.
|   m1<m
|   ⇒
|   (∀u:A.
|    M(u) = m1
|    ⇒
|    (∀j:B.
|     process u j  where
|     process s i ==
|     if P(i,s) then F(i,s)
|     else G(i,s) where
|          xs := N(i,s);
|          s:= H(i,s);
|          while not null xs {
|          s := process s (hd xs);
|          xs := tl xs;
|          }
|     ∈ {s:A| M(s)≤M(u) }))
14. u : A
15. M(u) = m
16. j : B
17. ¬P(j,u)
⊢  G(j
|   ,list_accum(s',i'.process s' i'  where
|   process s i ==
|   if P(i,s) then F(i,s)
|   else G(i,s) where
|        xs := N(i,s);
|        s:= H(i,s);
|        while not null xs {
|        s := process s (hd xs);
|        xs := tl xs;
|        }  ;H(j,u);N(j,u)))
|  ∈ {s:A| M(s)≤M(u) }
| by (GenConcl
|     (list_accum(s',i'.process s' i'  where
|      process s i ==
|      if P(i,s) then F(i,s)
|      else G(i,s) where
|           xs := N(i,s);
|           s:= H(i,s);
|           while not null xs {
|           s := process s (hd xs);
|           xs := tl xs;
|           }  ;H(j,u);N(j,u))
|      =
```

```
|      a))
|    THEN
|    (Try (Complete Auto))
|
|\
| .....wf .....
| 1. A : Type
| 2. B : Type
| 3. P : B→A→𝔹
| 4. F : B→A→A
| 5. G : B→A→A
| 6. H : B→A→A
| 7. N : B→A→(B List)
| 8. M : A→ℕ
| 9. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
| 10. ∀i:B, s:A. M(G(i,s))≤M(s)
| 11. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
| 12. m : ℕ
| 13. ∀m1:ℕ.
| |   m1<m
| |   ⇒
| |   (∀u:A.
| |    M(u) = m1
| |    ⇒
| |    (∀j:B.
| |     process u j  where
| |     process s i ==
| |     if P(i,s) then F(i,s)
| |     else G(i,s) where
| |          xs := N(i,s);
| |          s:= H(i,s);
| |          while not null xs {
| |          s := process s (hd xs);
| |          xs := tl xs;
| |          }
| |     ∈ {s:A| M(s)≤M(u) }))
| 14. u : A
| 15. M(u) = m
| 16. j : B
| 17. ¬P(j,u)
| ⊢  list_accum(s',i'.process s' i'  where
| |  process s i ==
| |  if P(i,s) then F(i,s)
| |  else G(i,s) where
| |       xs := N(i,s);
| |       s:= H(i,s);
| |       while not null xs {
| |       s := process s (hd xs);
| |       xs := tl xs;
| |       }  ;H(j,u);N(j,u))
| |  ∈ {s:A| M(s)≤M(H(j,u)) }
| | by (((GenConclAtAddr [2;3]) THEN (Thin -1)
| |      THEN
| |      (GenConcl (H(j,u) = a)))
| |    THENA
```

```
| |     (Auto THEN Analyze))
| |    THEN
| |    (Thin -1)
| |
| 1. A : Type
| 2. B : Type
| 3. P : B→A→𝔹
| 4. F : B→A→A
| 5. G : B→A→A
| 6. H : B→A→A
| 7. N : B→A→(B List)
| 8. M : A→ℕ
| 9. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
| 10. ∀i:B, s:A. M(G(i,s))≤M(s)
| 11. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
| 12. m : ℕ
| 13. ∀m1:ℕ.
| |   m1<m
| |   ⇒
| |   (∀u:A.
| |    M(u) = m1
| |    ⇒
| |    (∀j:B.
| |     process u j  where
| |     process s i ==
| |     if P(i,s) then F(i,s)
| |     else G(i,s) where
| |          xs := N(i,s);
| |          s:= H(i,s);
| |          while not null xs {
| |          s := process s (hd xs);
| |          xs := tl xs;
| |          }
| |     ∈ {s:A| M(s)≤M(u) }))
| 14. u : A
| 15. M(u) = m
| 16. j : B
| 17. ¬P(j,u)
| 18. z : B List
| 19. a : {s:A| M(s)≤M(H(j,u)) }
| ⊢  list_accum(s',i'.process s' i'  where
| |  process s i ==
| |  if P(i,s) then F(i,s)
| |  else G(i,s) where
| |       xs := N(i,s);
| |       s:= H(i,s);
| |       while not null xs {
| |       s := process s (hd xs);
| |       xs := tl xs;
| |       }  ;a;z)
| |  ∈ {s:A| M(s)≤M(a) }
| | by (MoveToConcl -1) THEN (ListInd -1) THEN (Reduce 0)
| |
| |\
| | .....subterm .....
```

```
| | 1. A : Type
| | 2. B : Type
| | 3. P : B→A→𝔹
| | 4. F : B→A→A
| | 5. G : B→A→A
| | 6. H : B→A→A
| | 7. N : B→A→(B List)
| | 8. M : A→ℕ
| | 9. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
| | 10. ∀i:B, s:A. M(G(i,s))≤M(s)
| | 11. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
| | 12. m : ℕ
| | 13. ∀m1:ℕ.
| |     m1<m
| |     ⇒
| |     (∀u:A.
| |      M(u) = m1
| |      ⇒
| |      (∀j:B.
| |       process u j  where
| |       process s i ==
| |       if P(i,s) then F(i,s)
| |       else G(i,s) where
| |            xs := N(i,s);
| |            s:= H(i,s);
| |            while not null xs {
| |            s := process s (hd xs);
| |            xs := tl xs;
| |            }
| |       ∈ {s:A| M(s)≤M(u) }))
| | 14. u : A
| | 15. M(u) = m
| | 16. j : B
| | 17. ¬P(j,u)
| | 18. B List
| | 19. u1 : B
| | 20. v : B List
| | 21. ∀a:{s:A| M(s)≤M(H(j,u)) }.
| |     list_accum(s',i'.process s' i'  where
| |     process s i ==
| |     if P(i,s) then F(i,s)
| |     else G(i,s) where
| |          xs := N(i,s);
| |          s:= H(i,s);
| |          while not null xs {
| |          s := process s (hd xs);
| |          xs := tl xs;
| |          }  ;a;v)
| |     ∈ {s:A| M(s)≤M(a) }
| | 22. a : {s:A| M(s)≤M(H(j,u)) }
| | ⊢  process a u1  where
| |    process s i ==
| |    if P(i,s) then F(i,s)
| |    else G(i,s) where
| |         xs := N(i,s);
```

```
| |         s:= H(i,s);
| |         while not null xs {
| |         s := process s (hd xs);
| |         xs := tl xs;
| |         }
| |    ∈ {s:A| M(s)≤M(a) }
| |   by (Using ['m1',M(a)] (BackThru 13))
| |      THEN
| |      (AssertBY (M(H(j,u))<M(u)) EasyHyp)
| |      THEN
| |      (Analyze -2)
| |
| .\
| . .....subterm .....
| . 1. A : Type
| . 2. B : Type
| . 3. P : B→A→𝔹
| . 4. F : B→A→A
| . 5. G : B→A→A
| . 6. H : B→A→A
| . 7. N : B→A→(B List)
| . 8. M : A→ℕ
| . 9. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
| . 10. ∀i:B, s:A. M(G(i,s))≤M(s)
| . 11. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
| . 12. m : ℕ
| . 13. ∀m1:ℕ.

| . |   m1<m
| . |   ⇒
| . |   (∀u:A.
| . |    M(u) = m1
| . |    ⇒
| . |    (∀j:B.
| . |     process u j  where
| . |     process s i ==
| . |     if P(i,s) then F(i,s)
| . |     else G(i,s) where
| . |          xs := N(i,s);
| . |          s:= H(i,s);
| . |          while not null xs {
| . |          s := process s (hd xs);
| . |          xs := tl xs;
| . |          }
| . |     ∈ {s:A| M(s)≤M(u) }))
| . 14. u : A
| . 15. M(u) = m
| . 16. j : B
| . 17. ¬P(j,u)
| . 18. B List
| . 19. B
| . 20. v : B List
| . 21. ∀a:{s:A| M(s)≤M(H(j,u)) }.
| . |   list_accum(s',i'.process s' i'  where
| . |   process s i ==
| . |   if P(i,s) then F(i,s)
```

```
| . |   else G(i,s) where
| . |        xs := N(i,s);
| . |        s:= H(i,s);
| . |        while not null xs {
| . |        s := process s (hd xs);
| . |        xs := tl xs;
| . |        }  ;a;v)
| . |   ∈ {s:A| M(s)≤M(a) }
| . 22. a : {s:A| M(s)≤M(H(j,u)) }
| . 23. s' : {s:A| M(s)≤M(a) }
| . 24. i' : B
| . ⊢  process s' i'  where
| . |  process s i ==
| . |  if P(i,s) then F(i,s)
| . |  else G(i,s) where
| . |       xs := N(i,s);
| . |       s:= H(i,s);
| . |       while not null xs {
| . |       s := process s (hd xs);
| . |       xs := tl xs;
| . |       }
| . |  ∈ {s:A| M(s)≤M(a) }  by InstHyp [M(s');s';i'] 13
| . |
| . |\
| . | .....antecedent .....
| . | 1. A : Type
| . | 2. B : Type
| . | 3. P : B→A→𝔹
| . | 4. F : B→A→A
| . | 5. G : B→A→A
| . | 6. H : B→A→A
| . | 7. N : B→A→(B List)
| . | 8. M : A→ℕ
| . | 9. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
| . | 10. ∀i:B, s:A. M(G(i,s))≤M(s)
| . | 11. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
| . | 12. m : ℕ
| . | 13. ∀m1:ℕ.
| . |     m1<m
| . |     ⇒
| . |     (∀u:A.
| . |      M(u) = m1
| . |      ⇒
| . |      (∀j:B.
| . |       process u j  where
| . |       process s i ==
| . |       if P(i,s) then F(i,s)
| . |       else G(i,s) where
| . |            xs := N(i,s);
| . |            s:= H(i,s);
| . |            while not null xs {
| . |            s := process s (hd xs);
| . |            xs := tl xs;
| . |            }
| . |       ∈ {s:A| M(s)≤M(u) }))
```

```
| . | 14. u : A
| . | 15. M(u) = m
| . | 16. j : B
| . | 17. ¬P(j,u)
| . | 18. B List
| . | 19. B
| . | 20. v : B List
| . | 21. ∀a:{s:A| M(s)≤M(H(j,u)) }.
| . |     list_accum(s',i'.process s' i'  where
| . |     process s i ==
| . |     if P(i,s) then F(i,s)
| . |     else G(i,s) where
| . |          xs := N(i,s);
| . |          s:= H(i,s);
| . |          while not null xs {
| . |          s := process s (hd xs);
| . |          xs := tl xs;
| . |          }  ;a;v)
| . |     ∈ {s:A| M(s)≤M(a) }
| . | 22. a : {s:A| M(s)≤M(H(j,u)) }
| . | 23. s' : {s:A| M(s)≤M(a) }
| . | 24. B
| . | ⊢  M(s')<m
| . |   by (Analyze -3) THEN (Analyze -2)
| . |      THEN
| . |      (AssertBY (M(H(j,u))<M(u)) EasyHyp)
| . |
| . .\
| . . 1. A : Type
| . . 2. B : Type
| . . 3. P : B→A→𝔹
| . . 4. F : B→A→A
| . . 5. G : B→A→A
| . . 6. H : B→A→A
| . . 7. N : B→A→(B List)
| . . 8. M : A→ℕ
| . . 9. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
| . . 10. ∀i:B, s:A. M(G(i,s))≤M(s)
| . . 11. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
| . . 12. m : ℕ
| . . 13. ∀m1:ℕ.
| . . |   m1<m
| . . |   ⇒
| . . |   (∀u:A.
| . . |    M(u) = m1
| . . |    ⇒
| . . |    (∀j:B.
| . . |     process u j  where
| . . |     process s i ==
| . . |     if P(i,s) then F(i,s)
| . . |     else G(i,s) where
| . . |          xs := N(i,s);
| . . |          s:= H(i,s);
| . . |          while not null xs {
| . . |          s := process s (hd xs);
```

```
| . . |          xs := tl xs;
| . . |          }
| . . |     ∈ {s:A| M(s)≤M(u) }))
| . . 14. u : A
| . . 15. M(u) = m
| . . 16. j : B
| . . 17. ¬P(j,u)
| . . 18. B List
| . . 19. B
| . . 20. v : B List
| . . 21. ∀a:{s:A| M(s)≤M(H(j,u)) }.
| . . |   list_accum(s',i'.process s' i'  where
| . . |   process s i ==
| . . |   if P(i,s) then F(i,s)
| . . |   else G(i,s) where
| . . |        xs := N(i,s);
| . . |        s:= H(i,s);
| . . |        while not null xs {
| . . |        s := process s (hd xs);
| . . |        xs := tl xs;
| . . |        }  ;a;v)
| . . |   ∈ {s:A| M(s)≤M(a) }
| . . 22. a : {s:A| M(s)≤M(H(j,u)) }
| . . 23. s' : {s:A| M(s)≤M(a) }
| . . 24. i' : B
| . . 25. process s' i'  where
| . . |   process s i ==
| . . |   if P(i,s) then F(i,s)
| . . |   else G(i,s) where
| . . |        xs := N(i,s);
| . . |        s:= H(i,s);
| . . |        while not null xs {
| . . |        s := process s (hd xs);
| . . |        xs := tl xs;
| . . |        }
| . . |   ∈ {s:A| M(s)≤M(s') }
| . . ⊢  process s' i'  where
| . . |  process s i ==
| . . |  if P(i,s) then F(i,s)
| . . |  else G(i,s) where
| . . |       xs := N(i,s);
| . . |       s:= H(i,s);
| . . |       while not null xs {
| . . |       s := process s (hd xs);
| . . |       xs := tl xs;
| . . |       }
| . . |  ∈ {s:A| M(s)≤M(a) }  by (GenConclAtAddr [2]) THEN (Thin -1)
| . . |
| . . 1. A : Type
| . . 2. B : Type
| . . 3. P : B→A→𝔹
| . . 4. F : B→A→A
| . . 5. G : B→A→A
| . . 6. H : B→A→A
| . . 7. N : B→A→(B List)
```

```
| . . 8. M : A→ℕ
| . . 9. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
| . . 10. ∀i:B, s:A. M(G(i,s))≤M(s)
| . . 11. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
| . . 12. m : ℕ
| . . 13. ∀m1:ℕ.
| . .     m1<m
| . .     ⇒
| . .     (∀u:A.
| . .      M(u) = m1
| . .      ⇒
| . .      (∀j:B.
| . .       process u j  where
| . .       process s i ==
| . .       if P(i,s) then F(i,s)
| . .       else G(i,s) where
| . .            xs := N(i,s);
| . .            s:= H(i,s);
| . .            while not null xs {
| . .            s := process s (hd xs);
| . .            xs := tl xs;
| . .            }
| . .       ∈ {s:A| M(s)≤M(u) }))
| . . 14. u : A
| . . 15. M(u) = m
| . . 16. j : B
| . . 17. ¬P(j,u)
| . . 18. B List
| . . 19. B
| . . 20. v : B List
| . . 21. ∀a:{s:A| M(s)≤M(H(j,u)) }.
| . .     list_accum(s',i'.process s' i'  where
| . .     process s i ==
| . .     if P(i,s) then F(i,s)
| . .     else G(i,s) where
| . .          xs := N(i,s);
| . .          s:= H(i,s);
| . .          while not null xs {
| . .          s := process s (hd xs);
| . .          xs := tl xs;
| . .          }  ;a;v)
| . .     ∈ {s:A| M(s)≤M(a) }
| . . 22. a : {s:A| M(s)≤M(H(j,u)) }
| . . 23. s' : {s:A| M(s)≤M(a) }
| . . 24. i' : B
| . . 25. process s' i'  where
| . .     process s i ==
| . .     if P(i,s) then F(i,s)
| . .     else G(i,s) where
| . .          xs := N(i,s);
| . .          s:= H(i,s);
| . .          while not null xs {
| . .          s := process s (hd xs);
| . .          xs := tl xs;
| . .          }
```

```
| . .     ∈ {s:A| M(s)≤M(s’) }
| . . 26. z1 : {s:A| M(s)≤M(s’) }
| . . ⊢  z1 ∈ {s:A| M(s)≤M(a) }
| . .   by (Analyze -4) THEN (Analyze -1) THEN Analyze
|
.\
. 1. A : Type
. 2. B : Type
. 3. P : B→A→𝔹
. 4. F : B→A→A
. 5. G : B→A→A
. 6. H : B→A→A
. 7. N : B→A→(B List)
. 8. M : A→ℕ
. 9. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
. 10. ∀i:B, s:A. M(G(i,s))≤M(s)
. 11. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
. 12. m : ℕ
. 13. ∀m1:ℕ.
.     m1<m
.     ⇒
.     (∀u:A.
.      M(u) = m1
.      ⇒
.      (∀j:B.
.       process u j  where
.       process s i ==
.       if P(i,s) then F(i,s)
.       else G(i,s) where
.            xs := N(i,s);
.            s:= H(i,s);
.            while not null xs {
.            s := process s (hd xs);
.            xs := tl xs;
.            }
.       ∈ {s:A| M(s)≤M(u) }))
. 14. u : A
. 15. M(u) = m
. 16. j : B
. 17. ¬P(j,u)
. 18. a : {s:A| M(s)≤M(H(j,u)) }
. 19. list_accum(s’,i’.process s’ i’  where
.     process s i ==
.     if P(i,s) then F(i,s)
.     else G(i,s) where
.          xs := N(i,s);
.          s:= H(i,s);
.          while not null xs {
.          s := process s (hd xs);
.          xs := tl xs;
.          }  ;H(j,u);N(j,u))
.     =

.     a
. ⊢  G(j,a) ∈ {s:A| M(s)≤M(u) }
.   by (Thin -1) THEN (Analyze -1) THEN Analyze
```

```
.       THEN
.       (AssertBY (M(G(j,a))≤M(a)) EasyHyp)
.       THEN
.       (AssertBY (M(H(j,u))<M(u)) EasyHyp)
END
```