```
accumulate-induction1  THM

⊢  ∀A,B:Type, M:(A→ℕ), Q:(B→A→A→Prop), P:(B→A→𝔹)
|  , F,G,H:(B→A→A), N:(B→A→(B List)).
|  (∀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. P(j,u) ⇒ Q(j,u,F(j,u)))
|  ⇒
|  (∀j:B, u,z:A. ¬P(j,u) ⇒ Q(j,H(j,u),z) ⇒ Q(j,u,G(j,z)))
|  ⇒
|  (∀j:B, u:A. Q(j,u,u))
|  ⇒
|  (∀i,j:B, u,v,z:A. Q(i,u,v) ⇒ Q(j,v,z) ⇒ Q(j,u,z))
|  ⇒
|  (∀j:B, u:A.
|   Q(j
|    ,u
|    ,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;
|         }  ))
| by Auto THEN (RepeatFor 2 (MoveToConcl -1))
|    THEN
|    (Assert
|     (∀m:ℕ, u:A.
|      M(u) = m
|      ⇒
|      (∀j:B.
|       Q(j
|        ,u
|        ,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;
|             }  ))))
|    THEN
|    (Try (Complete (Auto THEN (Using [‘m’,M(u)] EasyHyp))))
|    THEN
|    CompleteInductionOnNat
|    THEN
```

```
|    (RecUnfold `accumulate` 0)
|    THEN
|    SplitOnConclITE
|    THEN
|    (Try (Complete (Auto THEN EasyHyp)))
|    THEN
|    (Inst Thm: accumulate_wf [A;B;P;F;G;H;N;M])
|    THEN
|    (Try Trivial)
|    THEN
|    (AssertBY
|     (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)) })
|     (Auto THEN (Try (Analyze THEN (Complete Auto)))
|      THEN
|      (InstHyp [i';s'] -3)
|      THEN
|      (GenConclAtAddr [2])
|      THEN
|      (Thin -1)
|      THEN
|      (Analyze -4)
|      THEN
|      (Analyze -1)))
|    THEN
|    (AllHyps
|     (λh.
|      (BackThru h) THEN (Try (Complete Auto))
|      THEN
|      (Try
|       ((GenConclAtAddr [2]) THEN (Thin -1) THEN (Analyze -1)
|        THEN
|        (Complete Auto)))))
|
1. A : Type
2. B : Type
3. M : A→ℕ
4. Q : B→A→A→Prop
5. P : B→A→𝔹
6. F : B→A→A
7. G : B→A→A
8. H : B→A→A
9. N : B→A→(B List)
10. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
11. ∀i:B, s:A. M(G(i,s))≤M(s)
12. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
```

```
13. ∀j:B, u:A. P(j,u) ⇒ Q(j,u,F(j,u))
14. ∀j:B, u,z:A. ¬P(j,u) ⇒ Q(j,H(j,u),z) ⇒ Q(j,u,G(j,z))
15. ∀j:B, u:A. Q(j,u,u)
16. ∀i,j:B, u,v,z:A. Q(i,u,v) ⇒ Q(j,v,z) ⇒ Q(j,u,z)
17. m : ℕ
18. ∀m1:ℕ.
|   m1<m
|   ⇒
|   (∀u:A.
|    M(u) = m1
|    ⇒
|    (∀j:B.
|     Q(j
|      ,u
|      ,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;
|           }  )))
19. u : A
20. M(u) = m
21. j : B
22. ¬P(j,u)
23. ∀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) }
24. 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)) }
⊢  Q(j
|   ,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;
|        }  ;H(j,u);N(j,u)))
| by (((GenConclAtAddr [4;3]) THEN (GenConcl (H(j,u) = a)))
|     THENA
|     (Auto THEN Analyze))
|    THEN
|    (Thin -1)
|    THEN
|    (AssertBY z ⊆ N(j,u)
|     (BackThru
|      Thm* ∀L1,L2:T List. L1 = L2 ⇒ L1 ⊆ L2    {sublist_weakening}))
|    THEN
|    (MoveToConcl -1)
|    THEN
|    (MoveToConcl -1)
|    THEN
|    (Thin -1)
|    THEN
|    (ListInd -1)
|    THEN
|    (Reduce 0)
|
|\
| 1. A : Type
| 2. B : Type
| 3. M : A→ℕ
| 4. Q : B→A→A→Prop
| 5. P : B→A→𝔹
| 6. F : B→A→A
| 7. G : B→A→A
| 8. H : B→A→A
| 9. N : B→A→(B List)
| 10. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
| 11. ∀i:B, s:A. M(G(i,s))≤M(s)
| 12. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
| 13. ∀j:B, u:A. P(j,u) ⇒ Q(j,u,F(j,u))
| 14. ∀j:B, u,z:A. ¬P(j,u) ⇒ Q(j,H(j,u),z) ⇒ Q(j,u,G(j,z))
| 15. ∀j:B, u:A. Q(j,u,u)
| 16. ∀i,j:B, u,v,z:A. Q(i,u,v) ⇒ Q(j,v,z) ⇒ Q(j,u,z)
| 17. m : ℕ
| 18. ∀m1:ℕ.
|     m1<m
|     ⇒
|     (∀u:A.
|      M(u) = m1
|      ⇒
|      (∀j:B.
|       Q(j
|         ,u
```

```
|        ,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;
|             }  )))
| 19. u : A
| 20. M(u) = m
| 21. j : B
| 22. ¬P(j,u)
| 23. ∀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) }
| 24. 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)) }
| 25. B List
| 26. a : {s:A| M(s)≤M(H(j,u)) }
| 27. nil ⊆ N(j,u)
| ⊢  Q(j,a,a)  by EasyHyp
|
.\
. 1. A : Type
. 2. B : Type
. 3. M : A→ℕ
. 4. Q : B→A→A→Prop
. 5. P : B→A→𝔹
. 6. F : B→A→A
. 7. G : B→A→A
. 8. H : B→A→A
. 9. N : B→A→(B List)
. 10. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
. 11. ∀i:B, s:A. M(G(i,s))≤M(s)
. 12. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
```

```
. 13. ∀j:B, u:A. P(j,u) ⇒ Q(j,u,F(j,u))
. 14. ∀j:B, u,z:A. ¬P(j,u) ⇒ Q(j,H(j,u),z) ⇒ Q(j,u,G(j,z))
. 15. ∀j:B, u:A. Q(j,u,u)
. 16. ∀i,j:B, u,v,z:A. Q(i,u,v) ⇒ Q(j,v,z) ⇒ Q(j,u,z)
. 17. m : ℕ
. 18. ∀m1:ℕ.
. |   m1<m
. |   ⇒
. |   (∀u:A.
. |    M(u) = m1
. |    ⇒
. |    (∀j:B.
. |     Q(j
. |      ,u
. |      ,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;
. |           }  )))
. 19. u : A
. 20. M(u) = m
. 21. j : B
. 22. ¬P(j,u)
. 23. ∀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) }
. 24. 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)) }
. 25. B List
. 26. u1 : B
. 27. v : B List
. 28. ∀a:{s:A| M(s)≤M(H(j,u)) }.
```

```
. |   v ⊆ N(j,u)
. |   ⇒
. |   Q(j
. |    ,a
. |    ,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))
. 29. a : {s:A| M(s)≤M(H(j,u)) }
. 30. [u1 / v] ⊆ N(j,u)
. ⊢  Q(j
. |   ,a
. |   ,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;
. |        }  ;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;
. |        }  ;v))
. | by AllHyps
. |    (λh.
. |     (InstHyp [M(a);a;u1] h)
. |     THENA
. |     (Auto THEN (Analyze -2)
. |      THEN
. |      (AssertBY (M(H(j,u))<M(u)) EasyHyp)
. |      THEN
. |      (Complete Auto)))
. |
. 1. A : Type
. 2. B : Type
. 3. M : A→ℕ
. 4. Q : B→A→A→Prop
. 5. P : B→A→𝔹
. 6. F : B→A→A
. 7. G : B→A→A
. 8. H : B→A→A
```

```
. 9. N : B→A→(B List)
. 10. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
. 11. ∀i:B, s:A. M(G(i,s))≤M(s)
. 12. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
. 13. ∀j:B, u:A. P(j,u) ⇒ Q(j,u,F(j,u))
. 14. ∀j:B, u,z:A. ¬P(j,u) ⇒ Q(j,H(j,u),z) ⇒ Q(j,u,G(j,z))
. 15. ∀j:B, u:A. Q(j,u,u)
. 16. ∀i,j:B, u,v,z:A. Q(i,u,v) ⇒ Q(j,v,z) ⇒ Q(j,u,z)
. 17. m : ℕ
. 18. ∀m1:ℕ.
. |   m1<m
. |   ⇒
. |   (∀u:A.
. |    M(u) = m1
. |    ⇒
. |    (∀j:B.
. |     Q(j
. |       ,u
. |       ,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;
. |            }  )))
. 19. u : A
. 20. M(u) = m
. 21. j : B
. 22. ¬P(j,u)
. 23. ∀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) }
. 24. 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)) }
```

```
. 25. B List
. 26. u1 : B
. 27. v : B List
. 28. ∀a:{s:A| M(s)≤M(H(j,u)) }.
. |   v ⊆ N(j,u)
. |   ⇒
. |   Q(j
. |    ,a
. |    ,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))
. 29. a : {s:A| M(s)≤M(H(j,u)) }
. 30. [u1 / v] ⊆ N(j,u)
. 31. Q(u1
. |    ,a
. |    ,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;
. |         }  )
. ⊢  Q(j
. |   ,a
. |   ,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;
. |        }  ;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;
. |        }  ;v))
. | by (AssertBY
. |     (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) })
. |     EasyHyp)
. |    THEN
. |    (Using
. |     [‘i’,u1
. |     ;‘v’
. |      ,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;
. |            }  ]
. |     (BackThru 16))
. |    THEN
. |    (Try (Complete ((GenConclAtAddr [2]) THEN (Analyze -2))))
. |
. |\
. | .....subterm .....
. | 1. A : Type
. | 2. B : Type
. | 3. M : A→ℕ
. | 4. Q : B→A→A→Prop
. | 5. P : B→A→𝔹
. | 6. F : B→A→A
. | 7. G : B→A→A
. | 8. H : B→A→A
. | 9. N : B→A→(B List)
. | 10. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
. | 11. ∀i:B, s:A. M(G(i,s))≤M(s)
. | 12. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
. | 13. ∀j:B, u:A. P(j,u) ⇒ Q(j,u,F(j,u))
. | 14. ∀j:B, u,z:A. ¬P(j,u) ⇒ Q(j,H(j,u),z) ⇒ Q(j,u,G(j,z))
. | 15. ∀j:B, u:A. Q(j,u,u)
. | 16. ∀i,j:B, u,v,z:A. Q(i,u,v) ⇒ Q(j,v,z) ⇒ Q(j,u,z)
. | 17. m : ℕ
. | 18. ∀m1:ℕ.
. |     m1<m
. |     ⇒
. |     (∀u:A.
. |      M(u) = m1
. |      ⇒
. |      (∀j:B.
```

```
. |       Q(j
. |        ,u
. |        ,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;
. |             }  )))
. | 19. u : A
. | 20. M(u) = m
. | 21. j : B
. | 22. ¬P(j,u)
. | 23. ∀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) }
. | 24. 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)) }
. | 25. B List
. | 26. u1 : B
. | 27. v : B List
. | 28. ∀a:{s:A| M(s)≤M(H(j,u)) }.
. |     v ⊆ N(j,u)
. |     ⇒
. |     Q(j
. |      ,a
. |      ,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))
. | 29. a : {s:A| M(s)≤M(H(j,u)) }
. | 30. [u1 / v] ⊆ N(j,u)
. | 31. Q(u1
. |      ,a
. |      ,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;
. |           }  )
. | 32. 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) }
. | 33. s' : A
. | 34. 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;
. |         }
. |    ∈ A
. |   by AllHyps
. |      (λh.
. |       (InstHyp [i';s'] h) THEN (GenConclAtAddr [2])
. |       THEN
. |       (Analyze -2)
. |       THEN
. |       (Complete Auto))
. |
. .\
. . 1. A : Type
. . 2. B : Type
. . 3. M : A→ℕ
. . 4. Q : B→A→A→Prop
. . 5. P : B→A→𝔹
. . 6. F : B→A→A
```

```
. . 7. G : B→A→A
. . 8. H : B→A→A
. . 9. N : B→A→(B List)
. . 10. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
. . 11. ∀i:B, s:A. M(G(i,s))≤M(s)
. . 12. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
. . 13. ∀j:B, u:A. P(j,u) ⇒ Q(j,u,F(j,u))
. . 14. ∀j:B, u,z:A. ¬P(j,u) ⇒ Q(j,H(j,u),z) ⇒ Q(j,u,G(j,z))
. . 15. ∀j:B, u:A. Q(j,u,u)
. . 16. ∀i,j:B, u,v,z:A. Q(i,u,v) ⇒ Q(j,v,z) ⇒ Q(j,u,z)
. . 17. m : ℕ
. . 18. ∀m1:ℕ.
. . |   m1<m
. . |   ⇒
. . |   (∀u:A.
. . |    M(u) = m1
. . |    ⇒
. . |    (∀j:B.
. . |     Q(j
. . |      ,u
. . |      ,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;
. . |           }  )))
. . 19. u : A
. . 20. M(u) = m
. . 21. j : B
. . 22. ¬P(j,u)
. . 23. ∀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) }
. . 24. 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)) }
. . 25. B List
. . 26. u1 : B
. . 27. v : B List
. . 28. ∀a:{s:A| M(s)≤M(H(j,u)) }.
. . |   v ⊆ N(j,u)
. . |   ⇒
. . |   Q(j
. . |    ,a
. . |    ,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))
. . 29. a : {s:A| M(s)≤M(H(j,u)) }
. . 30. [u1 / v] ⊆ N(j,u)
. . 31. Q(u1
. . |    ,a
. . |    ,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;
. . |         }  )
. . 32. 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) }
. . ⊢  Q(j
. . |   ,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;
```

```
. . |        }
. . |   ,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;
. . |        }  ;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;
. . |        }  ;v))  by BackThru -5
. . |
. . |\
. . | .....wf .....
. . | 1. A : Type
. . | 2. B : Type
. . | 3. M : A→ℕ
. . | 4. Q : B→A→A→Prop
. . | 5. P : B→A→𝔹
. . | 6. F : B→A→A
. . | 7. G : B→A→A
. . | 8. H : B→A→A
. . | 9. N : B→A→(B List)
. . | 10. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
. . | 11. ∀i:B, s:A. M(G(i,s))≤M(s)
. . | 12. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
. . | 13. ∀j:B, u:A. P(j,u) ⇒ Q(j,u,F(j,u))
. . | 14. ∀j:B, u,z:A. ¬P(j,u) ⇒ Q(j,H(j,u),z) ⇒ Q(j,u,G(j,z))
. . | 15. ∀j:B, u:A. Q(j,u,u)
. . | 16. ∀i,j:B, u,v,z:A. Q(i,u,v) ⇒ Q(j,v,z) ⇒ Q(j,u,z)
. . | 17. m : ℕ
. . | 18. ∀m1:ℕ.
. . |     m1<m
. . |     ⇒
. . |     (∀u:A.
. . |      M(u) = m1
. . |      ⇒
. . |      (∀j:B.
. . |       Q(j
. . |         ,u
. . |         ,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;
. . |             }  )))
. . | 19. u : A
. . | 20. M(u) = m
. . | 21. j : B
. . | 22. ¬P(j,u)
. . | 23. ∀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) }
. . | 24. 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)) }
. . | 25. B List
. . | 26. u1 : B
. . | 27. v : B List
. . | 28. ∀a:{s:A| M(s)≤M(H(j,u)) }.
. . |     v ⊆ N(j,u)
. . |     ⇒
. . |     Q(j
. . |      ,a
. . |      ,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))
. . | 29. a : {s:A| M(s)≤M(H(j,u)) }
. . | 30. [u1 / v] ⊆ N(j,u)
. . | 31. Q(u1
. . |      ,a
. . |      ,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;
. . |           }  )
. . | 32. 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) }
. . | ⊢  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(H(j,u)) }
. . |   by (GenConclAtAddr [2]) THEN (Thin -1) THEN Analyze
. . |      THEN
. . |      (Analyze -5)
. . |      THEN
. . |      (Analyze -1)
. . |
. . .\
. . . 1. A : Type
. . . 2. B : Type
. . . 3. M : A→ℕ
. . . 4. Q : B→A→A→Prop
. . . 5. P : B→A→𝔹
. . . 6. F : B→A→A
. . . 7. G : B→A→A
. . . 8. H : B→A→A
. . . 9. N : B→A→(B List)
. . . 10. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
. . . 11. ∀i:B, s:A. M(G(i,s))≤M(s)
. . . 12. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
. . . 13. ∀j:B, u:A. P(j,u) ⇒ Q(j,u,F(j,u))
. . . 14. ∀j:B, u,z:A. ¬P(j,u) ⇒ Q(j,H(j,u),z) ⇒ Q(j,u,G(j,z))
. . . 15. ∀j:B, u:A. Q(j,u,u)
. . . 16. ∀i,j:B, u,v,z:A. Q(i,u,v) ⇒ Q(j,v,z) ⇒ Q(j,u,z)
. . . 17. m : ℕ
. . . 18. ∀m1:ℕ.
```

```
. . . |   m1<m
. . . |   ⇒
. . . |   (∀u:A.
. . . |    M(u) = m1
. . . |    ⇒
. . . |    (∀j:B.
. . . |     Q(j
. . . |      ,u
. . . |      ,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;
. . . |           }  )))
. . . 19. u : A
. . . 20. M(u) = m
. . . 21. j : B
. . . 22. ¬P(j,u)
. . . 23. ∀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) }
. . . 24. 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)) }
. . . 25. B List
. . . 26. u1 : B
. . . 27. v : B List
. . . 28. ∀a:{s:A| M(s)≤M(H(j,u)) }.
. . . |   v ⊆ N(j,u)
. . . |   ⇒
. . . |   Q(j
. . . |    ,a
. . . |    ,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))
. . . 29. a : {s:A| M(s)≤M(H(j,u)) }
. . . 30. [u1 / v] ⊆ N(j,u)
. . . 31. Q(u1
. . . |    ,a
. . . |    ,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;
. . . |         }  )
. . . 32. 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) }
. . . ⊢  v ⊆ N(j,u)
. . . | by Using [‘L2’,[u1 / v]]
. . . |    (BackThru
. . . |     Thm* ∀L1,L2,L3:T List. L1 ⊆ L2 ⇒ L2 ⊆ L3 ⇒ L1 ⊆ L3
. . . |     {sublist_transitivity})
. . . |
. . . 1. A : Type
. . . 2. B : Type
. . . 3. M : A→ℕ
. . . 4. Q : B→A→A→Prop
. . . 5. P : B→A→𝔹
. . . 6. F : B→A→A
. . . 7. G : B→A→A
. . . 8. H : B→A→A
. . . 9. N : B→A→(B List)
. . . 10. ∀i:B, s:A. P(i,s) ⇒ M(F(i,s))≤M(s)
. . . 11. ∀i:B, s:A. M(G(i,s))≤M(s)
. . . 12. ∀i:B, s:A. ¬P(i,s) ⇒ M(H(i,s))<M(s)
. . . 13. ∀j:B, u:A. P(j,u) ⇒ Q(j,u,F(j,u))
. . . 14. ∀j:B, u,z:A. ¬P(j,u) ⇒ Q(j,H(j,u),z) ⇒ Q(j,u,G(j,z))
. . . 15. ∀j:B, u:A. Q(j,u,u)
. . . 16. ∀i,j:B, u,v,z:A. Q(i,u,v) ⇒ Q(j,v,z) ⇒ Q(j,u,z)
```

```
. . . 17. m : ℕ
. . . 18. ∀m1:ℕ.
. . .     m1<m
. . .     ⇒
. . .     (∀u:A.
. . .      M(u) = m1
. . .      ⇒
. . .      (∀j:B.
. . .       Q(j
. . .         ,u
. . .         ,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;
. . .              }  )))
. . . 19. u : A
. . . 20. M(u) = m
. . . 21. j : B
. . . 22. ¬P(j,u)
. . . 23. ∀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) }
. . . 24. 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)) }
. . . 25. B List
. . . 26. u1 : B
. . . 27. v : B List
. . . 28. ∀a:{s:A| M(s)≤M(H(j,u)) }.
. . .     v ⊆ N(j,u)
. . .     ⇒
. . .     Q(j
. . .      ,a
```

```
. . .      ,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))
. . . 29. a : {s:A| M(s)≤M(H(j,u)) }
. . . 30. [u1 / v] ⊆ N(j,u)
. . . 31. Q(u1
. . .      ,a
. . .      ,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;
. . .           }  )
. . . 32. 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) }
. . . ⊢  v ⊆ [u1 / v]
. . .   by (BackThru
. . .       Thm* ∀L1,L2:T List. ¬null(L2) ⇒ L1 ⊆ tl(L2) ⇒ L1 ⊆ L2
. . .       {sublist_tl})
. . .      THEN
. . .      (Reduce 0)
. . .      THEN
. . .      (Try
. . .       (BackThru
. . .        Thm* ∀L1,L2:T List. L1 = L2 ⇒ L1 ⊆ L2
. . .        {sublist_weakening}))
. . .      THEN
. . .      (Analyze 0)
END
```