accumulate-rel THM

```
⊢  ∀A,A',B:Type, R:(A→A'→Prop), P:(B→A→𝔹), P':(B→A'→𝔹)
   , F,G,H:(B→A→A), F',G',H':(B→A'→A'), N:(B→A→(B List))
   , N':(B→A'→(B List)), M:(A→ℕ), 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))
   ⇒
   (∀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, v:A'. R(u,v) ⇒ (P(j,u) ⟺ P'(j,v)))
   ⇒
   (∀j:B, u:A, v:A'. R(u,v) ⇒ P(j,u) ⇒ P'(j,v) ⇒ R(F(j,u),F'(j,v)))
   ⇒
   (∀j:B, u:A, v:A'.
    R(u,v) ⇒ ¬P(j,u) ⇒ ¬P'(j,v) ⇒ R(H(j,u),H'(j,v)))
   ⇒
   (∀j:B, u:A, v:A'. R(u,v) ⇒ R(G(j,u),G'(j,v)))
   ⇒
   (∀j:B, u:A, v:A'. R(u,v) ⇒ N(j,u) = N'(j,v))
   ⇒
   (∀j:B, u:A, v:A'.
    R(u,v)
    ⇒
    R(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;
          }
     ,process v 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 4 (MoveToConcl -1))
       THEN
       (Assert
        (∀m:ℕ, u:A.
```

```
  M(u) = m
  ⇒
  (∀j:B, v:A'.
   R(u,v)
   ⇒
   R(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;
         }
    ,process v 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
(Repeat SplitOnConclITE)
THEN
(AutoInst [j;u;v])
THEN
(Inst Thm: accumulate_wf [A;B;P;F;G;H;N;M])
THEN
(Try Trivial)
THEN
(Inst Thm: accumulate_wf [A';B;P';F';G';H';N';M'])
THEN
(Try Trivial)
THEN
BackThruSomeHyp
THEN
(AllHyps
 (λh.
  (InstHyp [i';s'] h) THEN (Analyze -1) THEN (Complete Auto)))
THEN
(Subst (N(j,u) = N'(j,v)) 0)
THEN
(Try (Complete (Auto THEN EasyHyp)))
THEN
(GenConcl (H(j,u) = a)))
```

```
      THENA
      (Auto THEN Analyze))
     THEN
     (GenConcl (H'(j,v) = a'))
     THEN
     (AssertBY R(a,a')
      ((SubstFor a 0) THEN (SubstFor a' 0) THEN EasyHyp))
     THEN
     (RepeatFor 2 ((MoveToConcl -1) THEN (Thin -1)))
     THEN
     (MoveToConcl -1)
     THEN
     (GenConcl (N'(j,v) = L))
     THEN
     (Thin -1)
     THEN
     (ListInd -1)
     THEN
     (Reduce 0)
     THEN
     BackThruSomeHyp
     THEN
     (Try (Complete Auto))
     THEN
     (AllHyps
      (λh.
       (InstHyp [u1;a] h) THEN (GenConclAtAddr [2]) THEN (Thin -1)
       THEN
       (Analyze -5)
       THEN
       (Analyze -1)
       THEN
       (Complete Auto)))
     THEN
     (AllHyps
      (λh.
       (InstHyp [u1;a'] h) THEN (GenConclAtAddr [2]) THEN (Thin -1)
       THEN
       (Analyze -1)
       THEN
       (Complete Auto)))
     THEN
     (Using [`m1',M(a)] BackThruSomeHyp)
     THEN
     (Try (Complete Auto))
     THEN
     (Analyze -3)
     THEN
     (AssertBY (M(H(j,u))<M(u)) EasyHyp)
END
```