```
rel_mng_2_wf  THM

⊢  ∀r:rel(), ds,da:Collection(dec()), de:sig(), rho:Decl
|  , st1:Collection(SimpleType), e:{[[de]] rho}, s,s':{[[ds]] rho}
|  , a:[[st1]] rho, tr:trace_env([[da]] rho).
|  trace_consistent_rel(rho;da;tr.proj;r)
|  ⇒
|  tc(r;ds;st1;de)
|  ⇒
|  rel_mng_2(r; rho; ds; st1; de; e; s; s'; a; tr) ∈ Prop
| by Unfold `rel_mng_2` 0 THEN UnivCD THENM MoveToConcl (-1)

|    THENA Auto
|
|\
| 1. r : rel()
| 2. ds : Collection(dec())
| 3. da : Collection(dec())
| 4. de : sig()
| 5. rho : Decl
| 6. st1 : Collection(SimpleType)
| 7. e : {[[de]] rho}
| 8. s : {[[ds]] rho}
| 9. s' : {[[ds]] rho}
| 10. a : [[st1]] rho
| 11. tr : trace_env([[da]] rho)
| 12. trace_consistent_rel(rho;da;tr.proj;r)
| ⊢  tc(r;ds;st1;de)
| |  ⇒
| |  list_accum(x,t.x
| |                 ([[t]] 1of(e) s s' a tr);[[r.name]] rho ... ;...)
| |  ∈ Prop
| | by Unfolds ``let tc`` 0 THEN Reduce 0 THEN GenConcl r.args = l
| |     THENA Auto
| |
| 1. r : rel()
| 2. ds : Collection(dec())
| 3. da : Collection(dec())
| 4. de : sig()
| 5. rho : Decl
| 6. st1 : Collection(SimpleType)
| 7. e : {[[de]] rho}
| 8. s : {[[ds]] rho}
| 9. s' : {[[ds]] rho}
| 10. a : [[st1]] rho
| 11. tr : trace_env([[da]] rho)
| 12. trace_consistent_rel(rho;da;tr.proj;r)
| 13. l : Term List
| 14. r.args = l
| ⊢  Case(r.name)
| |  Case eq(Q) =>
| |      ||l|| = 2
| |      & Q ∈ term_types(ds;st1;de;l[0])
| |        & Q ∈ term_types(ds;st1;de;l[1])
| |  Case R =>
```

```
| |      ||de.rel(R)|| = ||l||
| |      & (∀i:ℕ.
| |         i<||l|| ⇒ (de.rel(R))[i] ∈ term_types(ds;st1;de;l[i]))
| |  Default => False
| |  ⇒
| |  list_accum(x,t.x
| |                 ([[t]] 1of(e) s s' a tr);[[r.name]] rho 2of(e) ;l)
| |  ∈ Prop
| | by D 1 THEN D 1 THEN All Reduce THEN
|
| |    Folds ``relname_eq relname_other`` 0 THEN D 0
|
| |    THENM ExRepD THENA Auto
| |
| |\
| | 1. x : SimpleType
| | 2. r1 : Term List
| | 3. ds : Collection(dec())
| | 4. da : Collection(dec())
| | 5. de : sig()
| | 6. rho : Decl
| | 7. st1 : Collection(SimpleType)
| | 8. e : {[[de]] rho}
| | 9. s : {[[ds]] rho}
| | 10. s' : {[[ds]] rho}
| | 11. a : [[st1]] rho
| | 12. tr : trace_env([[da]] rho)
| | 13. trace_consistent_rel(rho;da;tr.proj;<inl(x),r1>)
| | 14. l : Term List
| | 15. r1 = l
| | 16. ||l|| = 2
| | 17. x ∈ term_types(ds;st1;de;l[0])
| | 18. x ∈ term_types(ds;st1;de;l[1])
| | ⊢  list_accum(x,t.x
| | |                 ([[t]] 1of(e) s s' a tr);[[...]] rho 2of(e) ;l)
| | |  ∈ Prop
| | | by Using [`ls',[x; x]; `da',da;`de',de
| | |    ] (BackThruLemma `rel_mng_2_lemma`) THENA Auto
| | |
| | |    THEN Try (D 7 THEN All Reduce THEN Trivial)
| | |
| | |    THEN Try (Unfold `relname_mng` 0 THEN Reduce 0 THEN Auto)
| | |
| | |    THEN Try (Reduce 0 THEN ExRepD THEN Auto THEN
| | |
| | |             CaseNat 0 `i' THEN Reduce 0 THEN Auto
| | |
| | |             THEN CaseNat 1 `i' THEN Reduce 0 THEN Auto)
| | |
| | |\
| | | .....wf .....
| | | 1. x : SimpleType
| | | 2. r1 : Term List
| | | 3. ds : Collection(dec())
| | | 4. da : Collection(dec())
```

```
| | | 5. de : sig()
| | | 6. rho : Decl
| | | 7. st1 : Collection(SimpleType)
| | | 8. e : {[[de]] rho}
| | | 9. {[[ds]] rho}
| | | 10. {[[ds]] rho}
| | | 11. [[st1]] rho
| | | 12. tr : trace_env([[da]] rho)
| | | 13. trace_consistent_rel(rho;da;tr.proj;<inl(x),r1>)
| | | 14. l : Term List
| | | 15. r1 = l
| | | 16. ||l|| = 2
| | | 17. x ∈ term_types(ds;st1;de;l[0])
| | | 18. x ∈ term_types(ds;st1;de;l[1])
| | | ⊢ 1of(e) ∈ {1of([[de]] rho)}
| | |   by AllHyps (\i. D i THEN All Reduce THEN Trivial)
| | |
| | |\
| | | 1. x : SimpleType
| | | 2. r1 : Term List
| | | 3. ds : Collection(dec())
| | | 4. da : Collection(dec())
| | | 5. de : sig()
| | | 6. rho : Decl
| | | 7. st1 : Collection(SimpleType)
| | | 8. {[[de]] rho}
| | | 9. {[[ds]] rho}
| | | 10. {[[ds]] rho}
| | | 11. [[st1]] rho
| | | 12. tr : trace_env([[da]] rho)
| | | 13. trace_consistent_rel(rho;da;tr.proj;<inl(x),r1>)
| | | 14. l : Term List
| | | 15. r1 = l
| | | 16. ||l|| = 2
| | | 17. x ∈ term_types(ds;st1;de;l[0])
| | | 18. x ∈ term_types(ds;st1;de;l[1])
| | | 19. i : ℕ||l||
| | | 20. i = 0
| | | 21. ¬i = 1
| | | ⊢ trace_consistent(rho;da;tr.proj;l[0])
| | |   by AllHyps (\h. Unfold `trace_consistent_rel` h THEN Reduce h
| | |
| | |               THEN MoveToHyp (-1) h
| | |
| | |                    THEN SubstFor r1 (-2) THENA Auto
| | |
| | |                    THEN BackThruHyp (-2) THEN Auto')
| | |
| | .\
| | . 1. x : SimpleType
| | . 2. r1 : Term List
| | . 3. ds : Collection(dec())
| | . 4. da : Collection(dec())
| | . 5. de : sig()
| | . 6. rho : Decl
```

```
| | . 7. st1 : Collection(SimpleType)
| | . 8. {[[de]] rho}
| | . 9. {[[ds]] rho}
| | . 10. {[[ds]] rho}
| | . 11. [[st1]] rho
| | . 12. tr : trace_env([[da]] rho)
| | . 13. trace_consistent_rel(rho;da;tr.proj;<inl(x),r1>)
| | . 14. l : Term List
| | . 15. r1 = l
| | . 16. ||l|| = 2
| | . 17. x ∈ term_types(ds;st1;de;l[0])
| | . 18. x ∈ term_types(ds;st1;de;l[1])
| | . 19. i : ℕ||l||
| | . 20. ¬i = 0
| | . 21. i = 1
| | . ⊢  trace_consistent(rho;da;tr.proj;l[1])
| | .   by AllHyps (\h. Unfold `trace_consistent_rel` h THEN Reduce h
| | .
| | .               THEN MoveToHyp (-1) h
| | .
| | .                    THEN SubstFor r1 (-2) THENA Auto
| | .
| | .                    THEN BackThruHyp (-2) THEN Auto')
| |
| |\
| | 1. y : Label
| | 2. r1 : Term List
| | 3. ds : Collection(dec())
| | 4. da : Collection(dec())
| | 5. de : sig()
| | 6. rho : Decl
| | 7. st1 : Collection(SimpleType)
| | 8. e : {[[de]] rho}
| | 9. s : {[[ds]] rho}
| | 10. s' : {[[ds]] rho}
| | 11. a : [[st1]] rho
| | 12. tr : trace_env([[da]] rho)
| | 13. trace_consistent_rel(rho;da;tr.proj;<inr(y),r1>)
| | 14. l : Term List
| | 15. r1 = l
| | 16. ||de.rel(y)|| = ||l||
| | 17. ∀i:ℕ. i<||l|| ⇒ (de.rel(y))[i] ∈ term_types(ds;st1;de;l[i])
| | ⊢  list_accum(x,t.x
| | |                ([[t]] 1of(e) s s' a tr);[[...]] rho 2of(e) ;l)
| | |  ∈ Prop  by Unfold `relname_mng` 0 THEN Reduce 0
| | |
| | 1. y : Label
| | 2. r1 : Term List
| | 3. ds : Collection(dec())
| | 4. da : Collection(dec())
| | 5. de : sig()
| | 6. rho : Decl
| | 7. st1 : Collection(SimpleType)
| | 8. e : {[[de]] rho}
| | 9. s : {[[ds]] rho}
```

```
| | 10. s' : {[[ds]] rho}
| | 11. a : [[st1]] rho
| | 12. tr : trace_env([[da]] rho)
| | 13. trace_consistent_rel(rho;da;tr.proj;<inr(y),r1>)
| | 14. l : Term List
| | 15. r1 = l
| | 16. ||de.rel(y)|| = ||l||
| | 17. ∀i:ℕ. i<||l|| ⇒ (de.rel(y))[i] ∈ term_types(ds;st1;de;l[i])
| | ⊢  list_accum(x,t.x([[t]] 1of(e) s s' a tr);2of(e).y;l) ∈ Prop
| | | by let i = 8 in
| | let j = 9 in
| | D i THEN All Reduce THEN
| |
| | |    Unfold `sig_mng` j THEN
| | Unfold `record` j THEN
| |
| | |    Unfold `decl_type` j THEN
| | Reduce j THEN
| |
| | |    Unfold `st_list_mng` j
| | |
| | 1. y : Label
| | 2. r1 : Term List
| | 3. ds : Collection(dec())
| | 4. da : Collection(dec())
| | 5. de : sig()
| | 6. rho : Decl
| | 7. st1 : Collection(SimpleType)
| | 8. e1 : {1of([[de]] rho)}
| | 9. e2 : l:Label→reduce(λs,m. [[s]] rho→m;Prop;de.rel(l))
| | 10. s : {[[ds]] rho}
| | 11. s' : {[[ds]] rho}
| | 12. a : [[st1]] rho
| | 13. tr : trace_env([[da]] rho)
| | 14. trace_consistent_rel(rho;da;tr.proj;<inr(y),r1>)
| | 15. l : Term List
| | 16. r1 = l
| | 17. ||de.rel(y)|| = ||l||
| | 18. ∀i:ℕ. i<||l|| ⇒ (de.rel(y))[i] ∈ term_types(ds;st1;de;l[i])
| | ⊢  list_accum(x,t.x([[t]] e1 s s' a tr);e2.y;l) ∈ Prop
| | | by GenConcl e2.y = f THENA Auto
| | |
| | |\
| | | .....wf .....
| | | 1. y : Label
| | | 2. r1 : Term List
| | | 3. ds : Collection(dec())
| | | 4. da : Collection(dec())
| | | 5. de : sig()
| | | 6. rho : Decl
| | | 7. st1 : Collection(SimpleType)
| | | 8. {1of([[de]] rho)}
| | | 9. e2 : l:Label→reduce(λs,m. [[s]] rho→m;Prop;de.rel(l))
| | | 10. {[[ds]] rho}
| | | 11. {[[ds]] rho}
```

```
| | | 12. [[st1]] rho
| | | 13. tr : trace_env([[da]] rho)
| | | 14. trace_consistent_rel(rho;da;tr.proj;<inr(y),r1>)
| | | 15. l : Term List
| | | 16. r1 = l
| | | 17. ||de.rel(y)|| = ||l||
| | | 18. ∀i:ℕ. i<||l|| ⇒ (de.rel(y))[i] ∈ term_types(ds;st1;de;l[i])
| | | ⊢  e2.y ∈ reduce(λs,m. [[s]] rho→m;Prop;de.rel(y))
| | |   by Unfold `r_select` 0 THEN Auto
| | |
| | .\
| | . 1. y : Label
| | . 2. r1 : Term List
| | . 3. ds : Collection(dec())
| | . 4. da : Collection(dec())
| | . 5. de : sig()
| | . 6. rho : Decl
| | . 7. st1 : Collection(SimpleType)
| | . 8. e1 : {1of([[de]] rho)}
| | . 9. e2 : l:Label→reduce(λs,m. [[s]] rho→m;Prop;de.rel(l))
| | . 10. s : {[[ds]] rho}
| | . 11. s' : {[[ds]] rho}
| | . 12. a : [[st1]] rho
| | . 13. tr : trace_env([[da]] rho)
| | . 14. trace_consistent_rel(rho;da;tr.proj;<inr(y),r1>)
| | . 15. l : Term List
| | . 16. r1 = l
| | . 17. ||de.rel(y)|| = ||l||
| | . 18. ∀i:ℕ. i<||l|| ⇒ (de.rel(y))[i] ∈ term_types(ds;st1;de;l[i])
| | . 19. f : reduce(λs,m. [[s]] rho→m;Prop;de.rel(y))
| | . 20. e2.y = f
| | . ⊢  list_accum(x,t.x([[t]] e1 s s' a tr);f;l) ∈ Prop
| | . | by BackThruLemma `rel_mng_2_lemma` THEN Auto THEN EasyHyp
| | . |
| | . 1. y : Label
| | . 2. r1 : Term List
| | . 3. ds : Collection(dec())
| | . 4. da : Collection(dec())
| | . 5. de : sig()
| | . 6. rho : Decl
| | . 7. st1 : Collection(SimpleType)
| | . 8. {1of([[de]] rho)}
| | . 9. e2 : l:Label→reduce(λs,m. [[s]] rho→m;Prop;de.rel(l))
| | . 10. {[[ds]] rho}
| | . 11. {[[ds]] rho}
| | . 12. [[st1]] rho
| | . 13. tr : trace_env([[da]] rho)
| | . 14. trace_consistent_rel(rho;da;tr.proj;<inr(y),r1>)
| | . 15. l : Term List
| | . 16. r1 = l
| | . 17. ||de.rel(y)|| = ||l||
| | . 18. ∀i:ℕ. i<||l|| ⇒ (de.rel(y))[i] ∈ term_types(ds;st1;de;l[i])
| | . 19. f : reduce(λs,m. [[s]] rho→m;Prop;de.rel(y))
| | . 20. e2.y = f
| | . 21. i : ℕ||l||
```

```
| | . ⊢  trace_consistent(rho;da;tr.proj;l[i])
| | .   by AllHyps (\h. Unfold ‘trace_consistent_rel‘ h THEN Reduce h
| | .
| | .               THEN MoveToHyp (-1) h
| | .
| | .                    THEN SubstFor r1 (-2) THENA Auto
| | .
| | .                    THEN BackThruHyp (-2) THEN Auto’)
| |
| |\
| | .....set predicate .....
| | 1. y : Label
| | 2. r1 : Term List
| | 3. ds : Collection(dec())
| | 4. da : Collection(dec())
| | 5. de : sig()
| | 6. rho : Decl
| | 7. st1 : Collection(SimpleType)
| | 8. {[[de]] rho}
| | 9. {[[ds]] rho}
| | 10. {[[ds]] rho}
| | 11. [[st1]] rho
| | 12. tr : trace_env([[da]] rho)
| | 13. trace_consistent_rel(rho;da;tr.proj;<inr(y),r1>)
| | 14. l : Term List
| | 15. r1 = l
| | ⊢  0≤||de.rel(y)||  by Auto
| |
| .\
| . .....set predicate .....
| . 1. y : Label
| . 2. r1 : Term List
| . 3. ds : Collection(dec())
| . 4. da : Collection(dec())
| . 5. de : sig()
| . 6. rho : Decl
| . 7. st1 : Collection(SimpleType)
| . 8. {[[de]] rho}
| . 9. {[[ds]] rho}
| . 10. {[[ds]] rho}
| . 11. [[st1]] rho
| . 12. tr : trace_env([[da]] rho)
| . 13. trace_consistent_rel(rho;da;tr.proj;<inr(y),r1>)
| . 14. l : Term List
| . 15. r1 = l
| . ⊢  0≤||l||  by Auto
|
.\
. .....unproved Inclusion subgoal .....
. 1. rel()
. 2. Collection(dec())
. 3. Collection(dec())
. 4. de : sig()
. 5. rho : Decl
. 6. Collection(SimpleType)
```

```
. 7. [[de]] rho = [[de]] rho
. 8. Decl×Decl
. ⊢  Decl ⊆ Decl  by Unfold ‘decl‘ 0 THEN Auto
END
```