```
rel_mng_tappend  THM

⊢  ∀d:Decl, tr:trace_env(d), a:(Σd), r:rel(), rho,ds,da,de,e,s,v:Top.
|  ¬affects_trace_rel(tr.proj;kind(a);r) ⇒
|  ([[r]] rho ds da de e s v tappend(tr;a) ∼ ...)
| by ...
|  Reduce 0 THEN Assert ∀rho,ds,da,de,e,s,v,x:Top.
|    ¬affects_trace_rel(tr.proj;kind(a);<name,r1>) ⇒
|    (list_accum(x,t.x
|                  ([[t]] 1of(e) s v tappend(tr;a));x;r1) ∼ ...)
|
|\
| .....assertion .....
| 1. d : Decl
| 2. tr : trace_env(d)
| 3. a : (Σd)
| 4. name : relname()
| 5. r1 : Term List
| ⊢  ∀rho,ds,da,de,e,s,v,x:Top.
| |  ¬affects_trace_rel(tr.proj;kind(a);<name,r1>) ⇒
| |  (list_accum(x,t.x
| |                  ([[t]] 1of(e) s v tappend(tr;a));x;r1) ∼ ...)
| | by ListInd (-1) THEN All (Fold `mk_rel`)
|
| |       THEN All (Unfold `rel_mng`) THEN All Reduce
|
| |       THEN Try (Complete Auto)
| |
| 1. d : Decl
| 2. tr : trace_env(d)
| 3. a : (Σd)
| 4. name : relname()
| 5. Term List
| 6. u : Term
| 7. v : Term List
| 8. ∀rho,ds,da,de,e,s,v@0,x:Top.
| |  ¬affects_trace_rel(tr.proj;kind(a);mk_rel(name, v)) ⇒
| |  (list_accum(x,t.x
| |                  ([[t]] 1of(e) s v@0 tappend(tr;a));x;v) ∼ ...)
| ⊢  ∀rho,ds,da,de,e,s,v@0,x:Top.
| |  ¬affects_trace_rel(tr.proj;kind(a);mk_rel(name, [u / v])) ⇒
| |  ...
| | by ...
| |  THEN D 0 THENA Auto
| |
| 1. d : Decl
| 2. tr : trace_env(d)
| 3. a : (Σd)
| 4. name : relname()
| 5. Term List
| 6. u : Term
| 7. v : Term List
| 8. Top
| 9. Top
| 10. Top
```

```
| 11. Top
| 12. e : Top
| 13. s : Top
| 14. v@0 : Top
| 15. ∀x:Top.
| |   ¬affects_trace_rel(tr.proj;kind(a);mk_rel(name, v)) ⇒
| |   (list_accum(x,t.x
| |                  ([[t]] 1of(e) s v@0 tappend(tr;a));x;v) ∼ ...)
| 16. x : Top
| 17. ¬affects_trace_rel(tr.proj;kind(a);mk_rel(name, [u / v]))
| ⊢  list_accum(x,t.x
| |                ([[t]] 1of(e) s v@0 tappend(tr;a));x
| |                                                   (...);v) ∼ ...
| | by Subst [[u]] 1of(e) s v@0 tappend(tr;a) ∼ [[u]] 1of(e) s v@0 tr
| |    0
| |
| |\
| | .....equality .....
| | 1. d : Decl
| | 2. tr : trace_env(d)
| | 3. a : (Σd)
| | 4. name : relname()
| | 5. Term List
| | 6. u : Term
| | 7. v : Term List
| | 8. Top
| | 9. Top
| | 10. Top
| | 11. Top
| | 12. e : Top
| | 13. s : Top
| | 14. v@0 : Top
| | 15. ∀x:Top.
| | |   ¬affects_trace_rel(tr.proj;kind(a);mk_rel(name, v)) ⇒
| | |   (list_accum(x,t.x
| | |                  ([[t]] 1of(e) s v@0 tappend(tr;a));x;v) ∼ ...)
| | 16. Top
| | 17. ¬affects_trace_rel(tr.proj;kind(a);mk_rel(name, [u / v]))
| | ⊢  [[u]] 1of(e) s v@0 tappend(tr;a) ∼ [[u]] 1of(e) s v@0 tr
| | | by BackThruLemma ‘term_mng_tappend‘ THEN Auto
| | |
| | 1. d : Decl
| | 2. tr : trace_env(d)
| | 3. a : (Σd)
| | 4. name : relname()
| | 5. Term List
| | 6. u : Term
| | 7. v : Term List
| | 8. Top
| | 9. Top
| | 10. Top
| | 11. Top
| | 12. e : Top
| | 13. s : Top
| | 14. v@0 : Top
```

```
| | 15. ∀x:Top.
| | |   ¬affects_trace_rel(tr.proj;kind(a);mk_rel(name, v)) ⇒
| | |   (list_accum(x,t.x
| | |                   ([[t]] 1of(e) s v@0 tappend(tr;a));x;v) ∼ ...)
| | 16. Top
| | 17. ¬affects_trace_rel(tr.proj;kind(a);mk_rel(name, [u / v]))
| | ⊢  ¬affects_trace(tr.proj;kind(a);u)  by ParallelOp (-1)
| | |
| | 1. d : Decl
| | 2. tr : trace_env(d)
| | 3. a : (Σd)
| | 4. name : relname()
| | 5. Term List
| | 6. u : Term
| | 7. v : Term List
| | 8. Top
| | 9. Top
| | 10. Top
| | 11. Top
| | 12. e : Top
| | 13. s : Top
| | 14. v@0 : Top
| | 15. ∀x:Top.
| |     ¬affects_trace_rel(tr.proj;kind(a);mk_rel(name, v)) ⇒
| |     (list_accum(x,t.x
| |                   ([[t]] 1of(e) s v@0 tappend(tr;a));x;v) ∼ ...)
| | 16. Top
| | 17. affects_trace(tr.proj;kind(a);u)
| | ⊢  affects_trace_rel(tr.proj;kind(a);mk_rel(name, [u / v]))
| |   by All (Unfold `affects_trace_rel`) THEN All Reduce
| |
| |        THEN RW assert_pushdownC 0 THENA Auto THEN SimpConcl
| |
| .\
| . 1. d : Decl
| . 2. tr : trace_env(d)
| . 3. a : (Σd)
| . 4. name : relname()
| . 5. Term List
| . 6. u : Term
| . 7. v : Term List
| . 8. Top
| . 9. Top
| . 10. Top
| . 11. Top
| . 12. e : Top
| . 13. s : Top
| . 14. v@0 : Top
| . 15. ∀x:Top.
| . |   ¬affects_trace_rel(tr.proj;kind(a);mk_rel(name, v)) ⇒
| . |   (list_accum(x,t.x
| . |                   ([[t]] 1of(e) s v@0 tappend(tr;a));x;v) ∼ ...)
| . 16. x : Top
| . 17. ¬affects_trace_rel(tr.proj;kind(a);mk_rel(name, [u / v]))
| . ⊢  list_accum(x,t.x
```

```
| . |                 ([[t]] 1of(e) s v@0 tappend(tr;a));...;v) ~ ...
| . | by BackThruHyp (-3) THEN Auto
| . |
| . 1. d : Decl
| . 2. tr : trace_env(d)
| . 3. a : (Σd)
| . 4. name : relname()
| . 5. Term List
| . 6. u : Term
| . 7. v : Term List
| . 8. Top
| . 9. Top
| . 10. Top
| . 11. Top
| . 12. e : Top
| . 13. s : Top
| . 14. v@0 : Top
| . 15. ∀x:Top.
| . |   ¬affects_trace_rel(tr.proj;kind(a);mk_rel(name, v)) ⇒
| . |   (list_accum(x,t.x
| . |                   ([[t]] 1of(e) s v@0 tappend(tr;a));x;v) ~ ...)
| . 16. Top
| . 17. ¬affects_trace_rel(tr.proj;kind(a);mk_rel(name, [u / v]))
| . ⊢  ¬affects_trace_rel(tr.proj;kind(a);mk_rel(name, v))
| . | by ParallelOp (-1)
| . |
| . 1. d : Decl
| . 2. tr : trace_env(d)
| . 3. a : (Σd)
| . 4. name : relname()
| . 5. Term List
| . 6. u : Term
| . 7. v : Term List
| . 8. Top
| . 9. Top
| . 10. Top
| . 11. Top
| . 12. e : Top
| . 13. s : Top
| . 14. v@0 : Top
| . 15. ∀x:Top.
| .     ¬affects_trace_rel(tr.proj;kind(a);mk_rel(name, v)) ⇒
| .     (list_accum(x,t.x
| .                   ([[t]] 1of(e) s v@0 tappend(tr;a));x;v) ~ ...)
| . 16. Top
| . 17. affects_trace_rel(tr.proj;kind(a);mk_rel(name, v))
| . ⊢  affects_trace_rel(tr.proj;kind(a);mk_rel(name, [u / v]))
| .   by All (Unfold `affects_trace_rel`) THEN All Reduce
| .
| .        THEN RW assert_pushdownC 0 THENA Auto THEN SimpConcl
|
.\
. 1. d : Decl
. 2. tr : trace_env(d)
. 3. a : (Σd)
```

```
. 4. name : relname()
. 5. r1 : Term List
. 6. ∀rho,ds,da,de,e,s,v,x:Top.
.    ¬affects_trace_rel(tr.proj;kind(a);<name,r1>) ⇒
.    (list_accum(x,t.x
.                    ([[t]] 1of(e) s v tappend(tr;a));x;r1) ∼ ...)
. ⊢  ∀rho,ds,da,de,e,s,v:Top.
.    ¬affects_trace_rel(tr.proj;kind(a);<name,r1>) ⇒
.    (list_accum(x,t.x
.                    ([[t]] 1of(e) s v tappend(tr;a));...;r1) ∼ ...)
.   by Repeat (ParallelOp (-1) THEN Thin (-3))
.
.         THEN SimpleInstHyp [[name]] rho 2of(e)
.       (-1) THENA Auto THEN NthHyp (-1)
END
```