```
wp2_rel_correct  THM

⊢  ∀A:ioa{i:l}(), r:rel(), rho:Decl, de:sig(), e:{[[de]] rho}
|  , a:(Σ[[A.da]] rho), tr:trace_env([[A.da]] rho).
|  tc_ioa(A;de)
|  ⇒
|  ¬ioa_mentions_trace(A)
|  ⇒
|  trace_consistent_rel(rho;A.da;tr.proj;r)
|  ⇒
|  single_valued_decls(A.ds)
|  ⇒
|  (∀s,x':[[A]] rho de e.state.
|   tc(r;A.ds;dec_lookup(A.da;kind(a));de)
|   ⇒
|   covers_rel(A;r)
|   ⇒
|   [[A]] rho de e.trans(s,a,x')
|   ⇒
|   (rel_mng_2(r;
|              rho;
|              A.ds;
|              dec_lookup(A.da;kind(a));
|              de;
|              e;
|              s;
|              x';
|              value(a);
|              tr)
|    ⟺
|    [[wp2_rel(A;kind(a);r)]] rho A.ds ... de e s value(a) tr))
| by Auto

|       THEN Try (BackThruLemma ‘sigma_decls_mng_value‘ THEN Auto)

|       THEN TrivialIoaHyp
|  THEN Try (Reduce 0 THEN Auto)

|       THEN Try (Unfold ‘decl‘ 0 THEN Auto)

|       THEN Try (BackThruLemma ‘empty_sts_mng‘ THEN Auto)

|       THEN Try (Unfold ‘ioa_mng‘ 0 THEN Reduce 0 THEN Trivial)
|


|
|
|\
| 1. A : ioa{i:l}()
| 2. r : rel()
| 3. rho : Decl
| 4. de : sig()
| 5. e : {[[de]] rho}
| 6. a : (Σ[[A.da]] rho)
| 7. tr : trace_env([[A.da]] rho)
```

```
| 8. tc_ioa(A;de)
| 9. ¬ioa_mentions_trace(A)
| 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| 11. single_valued_decls(A.ds)
| 12. s : [[A]] rho de e.state
| 13. x' : [[A]] rho de e.state
| 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| 15. covers_rel(A;r)
| 16. [[A]] rho de e.trans(s,a,x')
| 17. rel_mng_2(r;
| |             rho;
| |             A.ds;
| |             dec_lookup(A.da;kind(a));
| |             de;
| |             e;
| |             s;
| |             x';
| |             value(a);
| |             tr)
| ⊢  [[wp2_rel(A;kind(a);r)]] rho A.ds ... de e s value(a) tr
| | by Unfold `ioa_mng` (-2) THEN Reduce (-2) THEN ExRepD THEN
|
| |    ...
| |
| 1. A : ioa{i:l}()
| 2. r : rel()
| 3. rho : Decl
| 4. de : sig()
| 5. e : {[[de]] rho}
| 6. a : (Σ[[A.da]] rho)
| 7. tr : trace_env([[A.da]] rho)
| 8. tc_ioa(A;de)
| 9. ¬ioa_mentions_trace(A)
| 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| 11. single_valued_decls(A.ds)
| 12. s : [[A]] rho de e.state
| 13. x' : [[A]] rho de e.state
| 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| 15. covers_rel(A;r)
| 16. ∀p:pre().
| |   p ∈ A.pre
| |   ⇒
| |   p.kind = kind(a)
| |   ⇒
| |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s value(a) ...
| 17. ∀ef:eff().
| |   ef ∈ A.eff
| |   ⇒
| |   ef.kind = kind(a)
| |   ⇒
| |   x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| 18. ∀fr:frame().
| |   fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| 19. rel_mng_2(r;
| |             rho;
```

```
| |             A.ds;
| |             dec_lookup(A.da;kind(a));
| |             de;
| |             e;
| |             s;
| |             x';
| |             value(a);
| |             tr)
| 20. r@0 : rel()
| 21. r@0 ∈ col_subst2(λx.
| |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| ⊢  [[r@0]] rho A.ds dec_lookup(A.da;kind(a)) de e s value(a) tr
| | by  RWW "member_col_subst2" (-1)
| THENA Auto
| THEN ExRepD
|
| |    THEN HypSubst' (-1) 0•
| |
| 1. A : ioa{i:l}()
| 2. r : rel()
| 3. rho : Decl
| 4. de : sig()
| 5. e : {[[de]] rho}
| 6. a : (Σ[[A.da]] rho)
| 7. tr : trace_env([[A.da]] rho)
| 8. tc_ioa(A;de)
| 9. ¬ioa_mentions_trace(A)
| 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| 11. single_valued_decls(A.ds)
| 12. s : [[A]] rho de e.state
| 13. x' : [[A]] rho de e.state
| 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| 15. covers_rel(A;r)
| 16. ∀p:pre().
| |   p ∈ A.pre
| |   ⇒
| |   p.kind = kind(a)
| |   ⇒
| |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s value(a) ...
| 17. ∀ef:eff().
| |   ef ∈ A.eff
| |   ⇒
| |   ef.kind = kind(a)
| |   ⇒
| |   x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| 18. ∀fr:frame().
| |   fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| 19. rel_mng_2(r;
| |             rho;
| |             A.ds;
| |             dec_lookup(A.da;kind(a));
| |             de;
| |             e;
| |             s;
| |             x';
```

```
| |             value(a);
| |             tr)
| 20. r@0 : rel()
| 21. as : (Label×Term) List
| 22. 1of(unzip(as)) = rel_primed_vars(r)
| 23. ∀i:ℕ.
| |   i<||as||
| |   ⇒
| |   2of(as[i]) ∈ (λx.
| |                 smts_eff(action_effect(kind(a);A.eff;A.frame);x))
| |                (1of(as[i]))
| 24. r@0 = rel_subst2(as;r)
| ⊢  [[rel_subst2(as;r)]] rho A.ds dec_lookup(A.da;...) de e s ... tr
| | by Assert value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho•
| |
| |\
| | .....assertion .....
| | 1. A : ioa{i:l}()
| | 2. r : rel()
| | 3. rho : Decl
| | 4. de : sig()
| | 5. e : {[[de]] rho}
| | 6. a : (Σ[[A.da]] rho)
| | 7. tr : trace_env([[A.da]] rho)
| | 8. tc_ioa(A;de)
| | 9. ¬ioa_mentions_trace(A)
| | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| | 11. single_valued_decls(A.ds)
| | 12. s : [[A]] rho de e.state
| | 13. x' : [[A]] rho de e.state
| | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| | 15. covers_rel(A;r)
| | 16. ∀p:pre().
| |     p ∈ A.pre
| |     ⇒
| |     p.kind = kind(a)
| |     ⇒
| |     [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| | 17. ∀ef:eff().
| |     ef ∈ A.eff
| |     ⇒
| |     ef.kind = kind(a)
| |     ⇒
| |     x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| | 18. ∀fr:frame().
| |     fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| | 19. rel_mng_2(r;
| |               rho;
| |               A.ds;
| |               dec_lookup(A.da;kind(a));
| |               de;
| |               e;
| |               s;
| |               x';
| |               value(a);
```

```
| |              tr)
| | 20. r@0 : rel()
| | 21. as : (Label×Term) List
| | 22. 1of(unzip(as)) = rel_primed_vars(r)
| | 23. ∀i:ℕ.
| |     i<||as||
| |     ⇒
| |     2of(as[i]) ∈ ...
| |                 (1of(as[i]))
| | 24. r@0 = rel_subst2(as;r)
| | ⊢  value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho  by Easy
| |
| .\
| . 1. A : ioa{i:l}()
| . 2. r : rel()
| . 3. rho : Decl
| . 4. de : sig()
| . 5. e : {[[de]] rho}
| . 6. a : (Σ[[A.da]] rho)
| . 7. tr : trace_env([[A.da]] rho)
| . 8. tc_ioa(A;de)
| . 9. ¬ioa_mentions_trace(A)
| . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . 11. single_valued_decls(A.ds)
| . 12. s : [[A]] rho de e.state
| . 13. x' : [[A]] rho de e.state
| . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . 15. covers_rel(A;r)
| . 16. ∀p:pre().
| . |   p ∈ A.pre
| . |   ⇒
| . |   p.kind = kind(a)
| . |   ⇒
| . |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . 17. ∀ef:eff().
| . |   ef ∈ A.eff
| . |   ⇒
| . |   ef.kind = kind(a)
| . |   ⇒
| . |   x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . 18. ∀fr:frame().
| . |   fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . 19. rel_mng_2(r;
| . |             rho;
| . |             A.ds;
| . |             dec_lookup(A.da;kind(a));
| . |             de;
| . |             e;
| . |             s;
| . |             x';
| . |             value(a);
| . |             tr)
| . 20. r@0 : rel()
| . 21. as : (Label×Term) List
| . 22. 1of(unzip(as)) = rel_primed_vars(r)
```

```
| . 23. ∀i:ℕ.
| . |   i<||as||
| . |   ⇒
| . |   2of(as[i]) ∈ ...
| . |                (1of(as[i]))
| . 24. r@0 = rel_subst2(as;r)
| . 25. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . ⊢  [[rel_subst2(as;r)]] rho A.ds dec_lookup(...;...) de e s ... tr
| . | by InstLemma ‘rel_subst_mng_2_iff‘ [r;as;A.ds;A.da;
| . |    dec_lookup(A.da;kind(a));de;rho;e;s;x';kind(a);value(a);tr
| . |    ] THENA Try (Complete Auto)
| . THEN TrivialIoaHyp•
| . |
| . |\
| . | 1. A : ioa{i:l}()
| . | 2. r : rel()
| . | 3. rho : Decl
| . | 4. de : sig()
| . | 5. e : {[[de]] rho}
| . | 6. a : (Σ[[A.da]] rho)
| . | 7. tr : trace_env([[A.da]] rho)
| . | 8. tc_ioa(A;de)
| . | 9. ¬ioa_mentions_trace(A)
| . | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | 11. single_valued_decls(A.ds)
| . | 12. s : [[A]] rho de e.state
| . | 13. x' : [[A]] rho de e.state
| . | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | 15. covers_rel(A;r)
| . | 16. ∀p:pre().
| . | |   p ∈ A.pre
| . | |   ⇒
| . | |   p.kind = kind(a)
| . | |   ⇒
| . | |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | 17. ∀ef:eff().
| . | |   ef ∈ A.eff
| . | |   ⇒
| . | |   ef.kind = kind(a)
| . | |   ⇒
| . | |   x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | 18. ∀fr:frame().
| . | |   fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | 19. rel_mng_2(r;
| . | |             rho;
| . | |             A.ds;
| . | |             dec_lookup(A.da;kind(a));
| . | |             de;
| . | |             e;
| . | |             s;
| . | |             x';
| . | |             value(a);
| . | |             tr)
| . | 20. r@0 : rel()
| . | 21. as : (Label×Term) List
```

```
| . | 22. 1of(unzip(as)) = rel_primed_vars(r)
| . | 23. ∀i:ℕ.
| . | |   i<||as||
| . | |   ⇒
| . | |   2of(as[i]) ∈ (λx.
| . | |                 smts_eff(action_effect(kind(a);A.eff;...);x))
| . | |               (1of(as[i]))
| . | 24. r@0 = rel_subst2(as;r)
| . | 25. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | ⊢  ¬subst_mentions_trace(as)
| . | | by ...
| . | THEN RepeatFor 2 (OrLeft THENA Auto)
| . | |
| . | 1. A : ioa{i:l}()
| . | 2. r : rel()
| . | 3. rho : Decl
| . | 4. de : sig()
| . | 5. e : {[[de]] rho}
| . | 6. a : (Σ[[A.da]] rho)
| . | 7. tr : trace_env([[A.da]] rho)
| . | 8. tc_ioa(A;de)
| . | 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | 10. single_valued_decls(A.ds)
| . | 11. s : [[A]] rho de e.state
| . | 12. x' : [[A]] rho de e.state
| . | 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | 14. covers_rel(A;r)
| . | 15. ∀p:pre().
| . | |   p ∈ A.pre
| . | |   ⇒
| . | |   p.kind = kind(a)
| . | |   ⇒
| . | |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | 16. ∀ef:eff().
| . | |   ef ∈ A.eff
| . | |   ⇒
| . | |   ef.kind = kind(a)
| . | |   ⇒
| . | |   x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | 17. ∀fr:frame().
| . | |   fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | 18. rel_mng_2(r;
| . | |             rho;
| . | |             A.ds;
| . | |             dec_lookup(A.da;kind(a));
| . | |             de;
| . | |             e;
| . | |             s;
| . | |             x';
| . | |             value(a);
| . | |             tr)
| . | 19. r@0 : rel()
| . | 20. as : (Label×Term) List
| . | 21. 1of(unzip(as)) = rel_primed_vars(r)
| . | 22. ∀i:ℕ.
```

```
| . | |   i<||as||
| . | |   ⇒
| . | |   2of(as[i]) ∈ (λx.
| . | |                  smts_eff(action_effect(kind(a);A.eff;...);x))
| . | |                 (1of(as[i]))
| . | 23. r@0 = rel_subst2(as;r)
| . | 24. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | 25. subst_mentions_trace(as)
| . | ⊢  ∃e:eff(). e ∈ A.eff & mentions_trace(e.smt.term)
| . | | by ...
| . | |
| . | 1. A : ioa{i:l}()
| . | 2. r : rel()
| . | 3. rho : Decl
| . | 4. de : sig()
| . | 5. e : {[[de]] rho}
| . | 6. a : (Σ[[A.da]] rho)
| . | 7. tr : trace_env([[A.da]] rho)
| . | 8. tc_ioa(A;de)
| . | 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | 10. single_valued_decls(A.ds)
| . | 11. s : [[A]] rho de e.state
| . | 12. x' : [[A]] rho de e.state
| . | 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | 14. covers_rel(A;r)
| . | 15. ∀p:pre().
| . | |   p ∈ A.pre
| . | |   ⇒
| . | |   p.kind = kind(a)
| . | |   ⇒
| . | |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | 16. ∀ef:eff().
| . | |   ef ∈ A.eff
| . | |   ⇒
| . | |   ef.kind = kind(a)
| . | |   ⇒
| . | |   x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | 17. ∀fr:frame().
| . | |   fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | 18. rel_mng_2(r;
| . | |             rho;
| . | |             A.ds;
| . | |             dec_lookup(A.da;kind(a));
| . | |             de;
| . | |             e;
| . | |             s;
| . | |             x';
| . | |             value(a);
| . | |             tr)
| . | 19. r@0 : rel()
| . | 20. as : (Label×Term) List
| . | 21. 1of(unzip(as)) = rel_primed_vars(r)
| . | 22. ∀i:ℕ.
| . | |   i<||as||
| . | |   ⇒
```

```
| . | |   2of(as[i]) ∈ (λx.
| . | |                 smts_eff(action_effect(kind(a);A.eff;...);x))
| . | |               (1of(as[i]))
| . | 23. r@0 = rel_subst2(as;r)
| . | 24. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | 25. i : ℕ||as||
| . | 26. mentions_trace(2of(as[i]))
| . | ⊢  ∃e:eff(). e ∈ A.eff & mentions_trace(e.smt.term)
| . | | by InstHyp [i] (-5) THENA Auto THEN Reduce (-1)
| . | |
| . | 1. A : ioa{i:l}()
| . | 2. r : rel()
| . | 3. rho : Decl
| . | 4. de : sig()
| . | 5. e : {[[de]] rho}
| . | 6. a : (Σ[[A.da]] rho)
| . | 7. tr : trace_env([[A.da]] rho)
| . | 8. tc_ioa(A;de)
| . | 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | 10. single_valued_decls(A.ds)
| . | 11. s : [[A]] rho de e.state
| . | 12. x' : [[A]] rho de e.state
| . | 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | 14. covers_rel(A;r)
| . | 15. ∀p:pre().
| . | |   p ∈ A.pre
| . | |   ⇒
| . | |   p.kind = kind(a)
| . | |   ⇒
| . | |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | 16. ∀ef:eff().
| . | |   ef ∈ A.eff
| . | |   ⇒
| . | |   ef.kind = kind(a)
| . | |   ⇒
| . | |   x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | 17. ∀fr:frame().
| . | |   fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | 18. rel_mng_2(r;
| . | |             rho;
| . | |             A.ds;
| . | |             dec_lookup(A.da;kind(a));
| . | |             de;
| . | |             e;
| . | |             s;
| . | |             x';
| . | |             value(a);
| . | |             tr)
| . | 19. r@0 : rel()
| . | 20. as : (Label×Term) List
| . | 21. 1of(unzip(as)) = rel_primed_vars(r)
| . | 22. ∀i:ℕ.
| . | |   i<||as||
| . | |   ⇒
| . | |   2of(as[i]) ∈ (λx.
```

```
| . | |                 smts_eff(action_effect(kind(a);A.eff;...);x))
| . | |                (1of(as[i]))
| . | 23. r@0 = rel_subst2(as;r)
| . | 24. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | 25. i : ℕ||as||
| . | 26. mentions_trace(2of(as[i]))
| . | 27. 2of(as[i]) ∈ smts_eff(action_effect(kind(a);A.eff;...);...)
| . | ⊢  ∃e:eff(). e ∈ A.eff & mentions_trace(e.smt.term)
| . | | by Unfold ‘smts_eff‘ (-1) THEN Unfold ‘smt_terms‘ (-1)
| . |
| . | |    THEN RW ColMemberC (-1) THEN ExRepD
| . |
| . | |    THEN Unfold ‘action_effect‘ (-3)
| . | THEN RW ColMemberC (-3)
| . | |
| . | THEN D (-3) THEN ExRepD
| . | |
| . | |\
| . | | 1. A : ioa{i:l}()
| . | | 2. r : rel()
| . | | 3. rho : Decl
| . | | 4. de : sig()
| . | | 5. e : {[[de]] rho}
| . | | 6. a : (Σ[[A.da]] rho)
| . | | 7. tr : trace_env([[A.da]] rho)
| . | | 8. tc_ioa(A;de)
| . | | 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | | 10. single_valued_decls(A.ds)
| . | | 11. s : [[A]] rho de e.state
| . | | 12. x’ : [[A]] rho de e.state
| . | | 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | | 14. covers_rel(A;r)
| . | | 15. ∀p:pre().
| . | |     p ∈ A.pre
| . | |     ⇒
| . | |     p.kind = kind(a)
| . | |     ⇒
| . | |     [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | | 16. ∀ef:eff().
| . | |     ef ∈ A.eff
| . | |     ⇒
| . | |     ef.kind = kind(a)
| . | |     ⇒
| . | |     x’.ef.smt.lbl
| . | |     =
| . | |     [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | | 17. ∀fr:frame().
| . | |     fr ∈ A.frame
| . | |     ⇒
| . | |     ¬(kind(a) ∈ fr.acts) ⇒ x’.fr.var = s.fr.var
| . | | 18. rel_mng_2(r;
| . | |               rho;
| . | |               A.ds;
| . | |               dec_lookup(A.da;kind(a));
| . | |               de;
```

```
| . | |              e;
| . | |              s;
| . | |              x';
| . | |              value(a);
| . | |              tr)
| . | | 19. r@0 : rel()
| . | | 20. as : (Label×Term) List
| . | | 21. 1of(unzip(as)) = rel_primed_vars(r)
| . | | 22. ∀i:ℕ.
| . | |     i<||as||
| . | |     ⇒
| . | |     2of(as[i]) ∈ ...
| . | |                  (1of(as[i]))
| . | | 23. r@0 = rel_subst2(as;r)
| . | | 24. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | | 25. i : ℕ||as||
| . | | 26. mentions_trace(2of(as[i]))
| . | | 27. s1 : smt()
| . | | 28. e1 : eff()
| . | | 29. e1 ∈ A.eff
| . | | 30. e1.kind =1 kind(a)
| . | | 31. s1 = e1.smt
| . | | 32. s1.lbl =1 1of(as[i])
| . | | 33. 2of(as[i]) = s1.term
| . | | ⊢  ∃e:eff(). e ∈ A.eff & mentions_trace(e.smt.term)
| . | |   by InstConcl [e1] THEN Auto
| . | | THEN SubstFor e1.smt 0
| . | |
| . | |      THENM SubstFor s1.term 0
| . | | THEN Auto
| . | |
| . | .\
| . | . 1. A : ioa{i:l}()
| . | . 2. r : rel()
| . | . 3. rho : Decl
| . | . 4. de : sig()
| . | . 5. e : {[[de]] rho}
| . | . 6. a : (Σ[[A.da]] rho)
| . | . 7. tr : trace_env([[A.da]] rho)
| . | . 8. tc_ioa(A;de)
| . | . 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . 10. single_valued_decls(A.ds)
| . | . 11. s : [[A]] rho de e.state
| . | . 12. x' : [[A]] rho de e.state
| . | . 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . 14. covers_rel(A;r)
| . | . 15. ∀p:pre().
| . | . |   p ∈ A.pre
| . | . |   ⇒
| . | . |   p.kind = kind(a)
| . | . |   ⇒
| . | . |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | . 16. ∀ef:eff().
| . | . |   ef ∈ A.eff
| . | . |   ⇒
```

```
| . | . |   ef.kind = kind(a)
| . | . |   ⇒
| . | . |   x'.ef.smt.lbl
| . | . |   =
| . | . |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . 17. ∀fr:frame().
| . | . |   fr ∈ A.frame
| . | . |   ⇒
| . | . |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . 18. rel_mng_2(r;
| . | . |             rho;
| . | . |             A.ds;
| . | . |             dec_lookup(A.da;kind(a));
| . | . |             de;
| . | . |             e;
| . | . |             s;
| . | . |             x';
| . | . |             value(a);
| . | . |             tr)
| . | . 19. r@0 : rel()
| . | . 20. as : (Label×Term) List
| . | . 21. 1of(unzip(as)) = rel_primed_vars(r)
| . | . 22. ∀i:ℕ.
| . | . |   i<||as||
| . | . |   ⇒
| . | . |   2of(as[i]) ∈ ...
| . | . |               (1of(as[i]))
| . | . 23. r@0 = rel_subst2(as;r)
| . | . 24. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . 25. i : ℕ||as||
| . | . 26. mentions_trace(2of(as[i]))
| . | . 27. s1 : smt()
| . | . 28. f : frame()
| . | . 29. f ∈ A.frame
| . | . 30. ¬₂kind(a) ∈ f.acts
| . | . 31. s1 = mk_smt(f.var, f.var, f.typ)
| . | . 32. s1.lbl =₁ 1of(as[i])
| . | . 33. 2of(as[i]) = s1.term
| . | . ⊢  ∃e:eff(). e ∈ A.eff & mentions_trace(e.smt.term)
| . | . | by Assert ¬mentions_trace(2of(as[i])) THEN Auto

| . | .
| . | . |    THEN SubstFor 2of(as[i]) 0
| . | . THENM SubstFor s1 0 THEN Auto
| . | . |
| . | . |
| . | . 1. A : ioa{i:l}()
| . | . 2. r : rel()
| . | . 3. rho : Decl
| . | . 4. de : sig()
| . | . 5. e : {[[de]] rho}
| . | . 6. a : (Σ[[A.da]] rho)
| . | . 7. tr : trace_env([[A.da]] rho)
| . | . 8. tc_ioa(A;de)
| . | . 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . 10. single_valued_decls(A.ds)
```

```
| . | . 11. s : [[A]] rho de e.state
| . | . 12. x' : [[A]] rho de e.state
| . | . 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . 14. covers_rel(A;r)
| . | . 15. ∀p:pre().
| . | .     p ∈ A.pre
| . | .     ⇒
| . | .     p.kind = kind(a)
| . | .     ⇒
| . | .     [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | . 16. ∀ef:eff().
| . | .     ef ∈ A.eff
| . | .     ⇒
| . | .     ef.kind = kind(a)
| . | .     ⇒
| . | .     x'.ef.smt.lbl
| . | .     =
| . | .     [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . 17. ∀fr:frame().
| . | .     fr ∈ A.frame
| . | .     ⇒
| . | .     ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . 18. rel_mng_2(r;
| . | .               rho;
| . | .               A.ds;
| . | .               dec_lookup(A.da;kind(a));
| . | .               de;
| . | .               e;
| . | .               s;
| . | .               x';
| . | .               value(a);
| . | .               tr)
| . | . 19. r@0 : rel()
| . | . 20. as : (Label×Term) List
| . | . 21. 1of(unzip(as)) = rel_primed_vars(r)
| . | . 22. ∀i:ℕ.
| . | .     i<||as||
| . | .     ⇒
| . | .     2of(as[i]) ∈ ...
| . | .                  (1of(as[i]))
| . | . 23. r@0 = rel_subst2(as;r)
| . | . 24. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . 25. i : ℕ||as||
| . | . 26. mentions_trace(2of(as[i]))
| . | . 27. s1 : smt()
| . | . 28. f : frame()
| . | . 29. f ∈ A.frame
| . | . 30. ¬₂kind(a) ∈ f.acts
| . | . 31. s1 = mk_smt(f.var, f.var, f.typ)
| . | . 32. s1.lbl =₁ 1of(as[i])
| . | . 33. 2of(as[i]) = s1.term
| . | . ⊢  ¬mentions_trace(mk_smt(f.var, f.var, f.typ).term)
| . | .   by Reduce 0 THEN SimpConcl
| . |
| . |\
```

```
| . | 1. A : ioa{i:l}()
| . | 2. r : rel()
| . | 3. rho : Decl
| . | 4. de : sig()
| . | 5. e : {[[de]] rho}
| . | 6. a : (Σ[[A.da]] rho)
| . | 7. tr : trace_env([[A.da]] rho)
| . | 8. tc_ioa(A;de)
| . | 9. ¬ioa_mentions_trace(A)
| . | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | 11. single_valued_decls(A.ds)
| . | 12. s : [[A]] rho de e.state
| . | 13. x' : [[A]] rho de e.state
| . | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | 15. covers_rel(A;r)
| . | 16. ∀p:pre().
| . | |   p ∈ A.pre
| . | |   ⇒
| . | |   p.kind = kind(a)
| . | |   ⇒
| . | |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | 17. ∀ef:eff().
| . | |   ef ∈ A.eff
| . | |   ⇒
| . | |   ef.kind = kind(a)
| . | |   ⇒
| . | |   x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | 18. ∀fr:frame().
| . | |   fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | 19. rel_mng_2(r;
| . | |             rho;
| . | |             A.ds;
| . | |             dec_lookup(A.da;kind(a));
| . | |             de;
| . | |             e;
| . | |             s;
| . | |             x';
| . | |             value(a);
| . | |             tr)
| . | 20. r@0 : rel()
| . | 21. as : (Label×Term) List
| . | 22. 1of(unzip(as)) = rel_primed_vars(r)
| . | 23. ∀i:ℕ.
| . | |   i<||as||
| . | |   ⇒
| . | |   2of(as[i]) ∈ (λx.
| . | |                 smts_eff(action_effect(kind(a);A.eff;...);x))
| . | |                (1of(as[i]))
| . | 24. r@0 = rel_subst2(as;r)
| . | 25. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | ⊢  ∀x:Label.
| . | |  (x ∈ rel_primed_vars(r))
| . | |  ⇒
| . | |  (∀t:SimpleType.
| . | |   mk_dec(x, t) ∈ A.ds
```

```
| . | |   ⇒
| . | |   t ∈ term_types(A.ds;dec_lookup(A.da;kind(a));de;...)
| . | |   & x'.x = [[apply_alist(as;x;x)]] 1of(e) s value(a) tr)
| . | | by MoveToHyp (-10) (-1) THEN
| . |
| . | |       RevHypSubstSq (-3) 0 THEN D 0 THENM D 0 THENA Auto
| . |
| . | |    THEN InstLemma `apply_alist_member` [Term;as;x;x
| . | |    ] THENA Auto
| . | THEN Unfold `l_member` (-1) THEN Reduce (-1)
| . | |
| . | THEN ExRepDHyps
| . | THEN Subst
| . | |    apply_alist(as;x;x) ∼ 2of(as[i]) 0
| . | THENM Subst
| . | |    x ∼ 1of(as[i]) 0
| . |
| . | |    ...
| . | THEN AllHyps (\i. InstHyp [i] i THENA Complete Auto)
| . |
| . | |    THEN Reduce (-1)
| . | THEN MoveToConcl (-1)
| . | THEN GenConcl
| . | |    2of(as[i]) = xxx
| . | THENM GenConcl 1of(as[i]) = xx
| . |
| . | |    THENA Auto
| . | THEN D 0 THENA Auto•
| . | |
| . | 1. A : ioa{i:l}()
| . | 2. r : rel()
| . | 3. rho : Decl
| . | 4. de : sig()
| . | 5. e : {[[de]] rho}
| . | 6. a : (Σ[[A.da]] rho)
| . | 7. tr : trace_env([[A.da]] rho)
| . | 8. tc_ioa(A;de)
| . | 9. ¬ioa_mentions_trace(A)
| . | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | 11. single_valued_decls(A.ds)
| . | 12. s : [[A]] rho de e.state
| . | 13. x' : [[A]] rho de e.state
| . | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | 15. covers_rel(A;r)
| . | 16. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | 17. ∀p:pre().
| . | |   p ∈ A.pre
| . | |   ⇒
| . | |   p.kind = kind(a)
| . | |   ⇒
| . | |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | 18. ∀ef:eff().
| . | |   ef ∈ A.eff
| . | |   ⇒
| . | |   ef.kind = kind(a)
```

```
| . | |   ⇒
| . | |   x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | 19. ∀fr:frame().
| . | |   fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | 20. rel_mng_2(r;
| . | |             rho;
| . | |             A.ds;
| . | |             dec_lookup(A.da;kind(a));
| . | |             de;
| . | |             e;
| . | |             s;
| . | |             x';
| . | |             value(a);
| . | |             tr)
| . | 21. r@0 : rel()
| . | 22. as : (Label×Term) List
| . | 23. 1of(unzip(as)) = rel_primed_vars(r)
| . | 24. ∀i:ℕ.
| . | |   i<||as||
| . | |   ⇒
| . | |   2of(as[i]) ∈ (λx.
| . | |                 smts_eff(action_effect(kind(a);A.eff;...);x))
| . | |                (1of(as[i]))
| . | 25. r@0 = rel_subst2(as;r)
| . | 26. x : Label
| . | 27. (x ∈ 1of(unzip(as)))
| . | 28. i : ℕ
| . | 29. i<||as||
| . | 30. <x,apply_alist(as;x;x)> = as[i]
| . | 31. xxx : Term
| . | 32. 2of(as[i]) = xxx
| . | 33. xx : Label
| . | 34. 1of(as[i]) = xx
| . | 35. xxx ∈ smts_eff(action_effect(kind(a);A.eff;A.frame);xx)
| . | ⊢  ∀t:SimpleType.
| . | |  mk_dec(xx, t) ∈ A.ds
| . | |  ⇒
| . | |  t ∈ term_types(A.ds;dec_lookup(A.da;kind(a));de;xxx)
| . | |  & x'.xx = [[xxx]] 1of(e) s value(a) tr
| . | | by Unfolds ``smts_eff action_effect`` (-1)
| . |
| . | |    THEN Unfold `smt_terms` (-1)
| . | THEN RW ColMemberC (-1)
| . |
| . | |    THEN ExRepDHyps
| . | THEN HypSubstSq (-1) 0
| . | THEN Thin (-1)
| . |
| . | |    THEN EqLblFwd
| . | THEN RevHypSubstSq (-1) 0
| . | THEN Thin (-1)
| . |
| . | |    THEN Thin (-1)
| . | THEN D (-1)
| . | THEN ExRepDHyps
```

```
| . |
| . | |    THEN HypSubstSq (-1) 0
| . | THEN Reduce 0•
| . | |
| . | |\
| . | | 1. A : ioa{i:l}()
| . | | 2. r : rel()
| . | | 3. rho : Decl
| . | | 4. de : sig()
| . | | 5. e : {[[de]] rho}
| . | | 6. a : (Σ[[A.da]] rho)
| . | | 7. tr : trace_env([[A.da]] rho)
| . | | 8. tc_ioa(A;de)
| . | | 9. ¬ioa_mentions_trace(A)
| . | | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | | 11. single_valued_decls(A.ds)
| . | | 12. s : [[A]] rho de e.state
| . | | 13. x' : [[A]] rho de e.state
| . | | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | | 15. covers_rel(A;r)
| . | | 16. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | | 17. ∀p:pre().
| . | | |   p ∈ A.pre
| . | | |   ⇒
| . | | |   p.kind = kind(a)
| . | | |   ⇒
| . | | |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | | 18. ∀ef:eff().
| . | | |   ef ∈ A.eff
| . | | |   ⇒
| . | | |   ef.kind = kind(a)
| . | | |   ⇒
| . | | |   x'.ef.smt.lbl
| . | | |   =
| . | | |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | | 19. ∀fr:frame().
| . | | |   fr ∈ A.frame
| . | | |   ⇒
| . | | |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | | 20. rel_mng_2(r;
| . | | |             rho;
| . | | |             A.ds;
| . | | |             dec_lookup(A.da;kind(a));
| . | | |             de;
| . | | |             e;
| . | | |             s;
| . | | |             x';
| . | | |             value(a);
| . | | |             tr)
| . | | 21. r@0 : rel()
| . | | 22. as : (Label×Term) List
| . | | 23. 1of(unzip(as)) = rel_primed_vars(r)
| . | | 24. ∀i:ℕ.
| . | | |   i<||as||
| . | | |   ⇒
```

```
| . | | |   2of(as[i]) ∈ ...
| . | | |               (1of(as[i]))
| . | | 25. r@0 = rel_subst2(as;r)
| . | | 26. x : Label
| . | | 27. (x ∈ 1of(unzip(as)))
| . | | 28. i : ℕ
| . | | 29. i<||as||
| . | | 30. <x,apply_alist(as;x;x)> = as[i]
| . | | 31. xxx : Term
| . | | 32. 2of(as[i]) = xxx
| . | | 33. xx : Label
| . | | 34. 1of(as[i]) = xx
| . | | 35. s1 : smt()
| . | | 36. e1 : eff()
| . | | 37. e1 ∈ A.eff
| . | | 38. e1.kind =₁ kind(a)
| . | | 39. s1 = e1.smt
| . | | ⊢  ∀t:SimpleType.
| . | | |  mk_dec(e1.smt.lbl, t) ∈ A.ds
| . | | |  ⇒
| . | | |  t ∈ term_types(A.ds;dec_lookup(A.da;kind(a));de;....term)
| . | | |  & x'.e1.smt.lbl = [[e1.smt.term]] 1of(e) s value(a) tr
| . | | | by EqLblFwd THEN AllHyps (\i. InstHyp [e1
| . | | |    ] i THENA Complete Auto) THEN
| . | | Auto THEN
| . | |
| . | | |    ...
| . | |
| . | | |    ...e1] (i+1) THENA Auto) THEN
| . | | Try (Subst t ∼ e1.smt.typ
| . | | |     0
| . | |
| . | | |    ...
| . | | |                              THEN InstHyp [
| . | | |    e1.smt.lbl] i THENA Auto
| . | |
| . | | |    ...
| . | | |           THEN  Auto;
| . | | |           Auto])•
| . | | |
| . | | |\
| . | | | 1. A : ioa{i:l}()
| . | | | 2. r : rel()
| . | | | 3. rho : Decl
| . | | | 4. de : sig()
| . | | | 5. e : {[[de]] rho}
| . | | | 6. a : (Σ[[A.da]] rho)
| . | | | 7. tr : trace_env([[A.da]] rho)
| . | | | 8. tc_pred(A.init;A.ds;<>;de)
| . | | | 9. ∀p:pre().
| . | | |    p ∈ A.pre ⇒ tc(p.rel;A.ds;dec_lookup(A.da;p.kind);de)
| . | | | 10. ∀ef:eff().
| . | | |     ef ∈ A.eff
| . | | |     ⇒
| . | | |     mk_dec(ef.kind, ef.typ) ∈ A.da
```

```
| . | | |     & mk_dec(ef.smt.lbl, ef.smt.typ) ∈ A.ds
| . | | |     & ef.smt.typ ∈ term_types(A.ds;<ef.typ>;de;ef.smt.term)
| . | | | 11. ∀f:frame(). f ∈ A.frame ⇒ mk_dec(f.var, f.typ) ∈ A.ds
| . | | | 12. ¬ioa_mentions_trace(A)
| . | | | 13. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | | | 14. single_valued_decls(A.ds)
| . | | | 15. s : [[A]] rho de e.state
| . | | | 16. x' : [[A]] rho de e.state
| . | | | 17. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | | | 18. covers_rel(A;r)
| . | | | 19. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | | | 20. ∀p:pre().
| . | | |     p ∈ A.pre
| . | | |     ⇒
| . | | |     p.kind = kind(a)
| . | | |     ⇒
| . | | |     [[p.rel]] rho A.ds dec_lookup(A.da;...) de e s ... ...
| . | | | 21. ∀ef:eff().
| . | | |     ef ∈ A.eff
| . | | |     ⇒
| . | | |     ef.kind = kind(a)
| . | | |     ⇒
| . | | |     x'.ef.smt.lbl
| . | | |     =
| . | | |     [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | | | 22. ∀fr:frame().
| . | | |     fr ∈ A.frame
| . | | |     ⇒
| . | | |     ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | | | 23. rel_mng_2(r;
| . | | |               rho;
| . | | |               A.ds;
| . | | |               dec_lookup(A.da;kind(a));
| . | | |               de;
| . | | |               e;
| . | | |               s;
| . | | |               x';
| . | | |               value(a);
| . | | |               tr)
| . | | | 24. r@0 : rel()
| . | | | 25. as : (Label×Term) List
| . | | | 26. 1of(unzip(as)) = rel_primed_vars(r)
| . | | | 27. ∀i:ℕ.
| . | | |     i<||as||
| . | | |     ⇒
| . | | |     2of(as[i]) ∈ ...
| . | | |                  (1of(as[i]))
| . | | | 28. r@0 = rel_subst2(as;r)
| . | | | 29. x : Label
| . | | | 30. (x ∈ 1of(unzip(as)))
| . | | | 31. i : ℕ
| . | | | 32. i<||as||
| . | | | 33. <x,apply_alist(as;x;x)> = as[i]
| . | | | 34. xxx : Term
| . | | | 35. 2of(as[i]) = xxx
```

```
| . | | | 36. xx : Label
| . | | | 37. 1of(as[i]) = xx
| . | | | 38. s1 : smt()
| . | | | 39. e1 : eff()
| . | | | 40. e1 ∈ A.eff
| . | | | 41. e1.kind =1 kind(a)
| . | | | 42. s1 = e1.smt
| . | | | 43. e1.kind = kind(a)
| . | | | 44. x'.e1.smt.lbl
| . | | |     =
| . | | |     [[e1.smt.term]] 1of(e) s value(a) niltrace()
| . | | | 45. t : SimpleType
| . | | | 46. mk_dec(e1.smt.lbl, t) ∈ A.ds
| . | | | 47. mk_dec(e1.kind, e1.typ) ∈ A.da
| . | | | 48. mk_dec(e1.smt.lbl, e1.smt.typ) ∈ A.ds
| . | | | 49. e1.smt.typ ∈ term_types(A.ds;<e1.typ>;de;e1.smt.term)
| . | | | ⊢  e1.smt.typ ∈ term_types(A.ds;dec_lookup(A.da;...);de;...)
| . | | |   by Using [‘ds1’,A.ds; ‘da1’,<e1.typ>
| . | | |      ...
| . | | |
| . | | |      ...
| . | | |
| . | | |      THEN Try (BackThruLemma ‘member_dec_lookup‘ THENA Auto)
| . | | |
| . | | | THEN Try (SubstFor kind(a) 0 THENM SubstFor x1
| . | | |        0 THEN Auto)•
| . | | |
| . | | .\
| . | | . 1. A : ioa{i:l}()
| . | | . 2. r : rel()
| . | | . 3. rho : Decl
| . | | . 4. de : sig()
| . | | . 5. e : {[[de]] rho}
| . | | . 6. a : (Σ[[A.da]] rho)
| . | | . 7. tr : trace_env([[A.da]] rho)
| . | | . 8. tc_pred(A.init;A.ds;<>;de)
| . | | . 9. ∀p:pre().
| . | | . |  p ∈ A.pre ⇒ tc(p.rel;A.ds;dec_lookup(A.da;p.kind);de)
| . | | . 10. ∀ef:eff().
| . | | . |   ef ∈ A.eff
| . | | . |   ⇒
| . | | . |   mk_dec(ef.kind, ef.typ) ∈ A.da
| . | | . |   & mk_dec(ef.smt.lbl, ef.smt.typ) ∈ A.ds
| . | | . |   & ef.smt.typ ∈ term_types(A.ds;<ef.typ>;de;ef.smt.term)
| . | | . 11. ∀f:frame(). f ∈ A.frame ⇒ mk_dec(f.var, f.typ) ∈ A.ds
| . | | . 12. ¬ioa_mentions_trace(A)
| . | | . 13. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | | . 14. single_valued_decls(A.ds)
| . | | . 15. s : [[A]] rho de e.state
| . | | . 16. x' : [[A]] rho de e.state
| . | | . 17. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | | . 18. covers_rel(A;r)
| . | | . 19. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | | . 20. ∀p:pre().
| . | | . |   p ∈ A.pre
```

```
| . | | . |   ⇒
| . | | . |   p.kind = kind(a)
| . | | . |   ⇒
| . | | . |   [[p.rel]] rho A.ds dec_lookup(A.da;...) de e s ... ...
| . | | . 21. ∀ef:eff().
| . | | . |   ef ∈ A.eff
| . | | . |   ⇒
| . | | . |   ef.kind = kind(a)
| . | | . |   ⇒
| . | | . |   x'.ef.smt.lbl
| . | | . |   =
| . | | . |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | | . 22. ∀fr:frame().
| . | | . |   fr ∈ A.frame
| . | | . |   ⇒
| . | | . |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | | . 23. rel_mng_2(r;
| . | | . |             rho;
| . | | . |             A.ds;
| . | | . |             dec_lookup(A.da;kind(a));
| . | | . |             de;
| . | | . |             e;
| . | | . |             s;
| . | | . |             x';
| . | | . |             value(a);
| . | | . |             tr)
| . | | . 24. r@0 : rel()
| . | | . 25. as : (Label×Term) List
| . | | . 26. 1of(unzip(as)) = rel_primed_vars(r)
| . | | . 27. ∀i:ℕ.
| . | | . |   i<||as||
| . | | . |   ⇒
| . | | . |   2of(as[i]) ∈ ...
| . | | . |               (1of(as[i]))
| . | | . 28. r@0 = rel_subst2(as;r)
| . | | . 29. x : Label
| . | | . 30. (x ∈ 1of(unzip(as)))
| . | | . 31. i : ℕ
| . | | . 32. i<||as||
| . | | . 33. <x,apply_alist(as;x;x)> = as[i]
| . | | . 34. xxx : Term
| . | | . 35. 2of(as[i]) = xxx
| . | | . 36. xx : Label
| . | | . 37. 1of(as[i]) = xx
| . | | . 38. s1 : smt()
| . | | . 39. e1 : eff()
| . | | . 40. e1 ∈ A.eff
| . | | . 41. e1.kind =1 kind(a)
| . | | . 42. s1 = e1.smt
| . | | . 43. e1.kind = kind(a)
| . | | . 44. x'.e1.smt.lbl
| . | | . |   =
| . | | . |   [[e1.smt.term]] 1of(e) s value(a) niltrace()
| . | | . 45. t : SimpleType
| . | | . 46. mk_dec(e1.smt.lbl, t) ∈ A.ds
```

```
| . | | . 47. mk_dec(e1.kind, e1.typ) ∈ A.da
| . | | . 48. mk_dec(e1.smt.lbl, e1.smt.typ) ∈ A.ds
| . | | . 49. e1.smt.typ ∈ term_types(A.ds;<e1.typ>;de;e1.smt.term)
| . | | . ⊢  x'.e1.smt.lbl = [[e1.smt.term]] 1of(e) s value(a) tr
| . | | . | by Subst
| . | | . |    [[e1.smt.term]] 1of(e) s value(a) tr ∼ ... 0
| . | | .
| . | | . |       THEN Try Trivial
| . | | . |
| . | | . .....equality .....
| . | | . 1. A : ioa{i:l}()
| . | | . 2. r : rel()
| . | | . 3. rho : Decl
| . | | . 4. de : sig()
| . | | . 5. e : {[[de]] rho}
| . | | . 6. a : (Σ[[A.da]] rho)
| . | | . 7. tr : trace_env([[A.da]] rho)
| . | | . 8. tc_pred(A.init;A.ds;<>;de)
| . | | . 9. ∀p:pre().
| . | | .    p ∈ A.pre ⇒ tc(p.rel;A.ds;dec_lookup(A.da;p.kind);de)
| . | | . 10. ∀ef:eff().
| . | | .     ef ∈ A.eff
| . | | .     ⇒
| . | | .     mk_dec(ef.kind, ef.typ) ∈ A.da
| . | | .     & mk_dec(ef.smt.lbl, ef.smt.typ) ∈ A.ds
| . | | .     & ef.smt.typ ∈ term_types(A.ds;<ef.typ>;de;ef.smt.term)
| . | | . 11. ∀f:frame(). f ∈ A.frame ⇒ mk_dec(f.var, f.typ) ∈ A.ds
| . | | . 12. ¬ioa_mentions_trace(A)
| . | | . 13. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | | . 14. single_valued_decls(A.ds)
| . | | . 15. s : [[A]] rho de e.state
| . | | . 16. x' : [[A]] rho de e.state
| . | | . 17. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | | . 18. covers_rel(A;r)
| . | | . 19. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | | . 20. ∀p:pre().
| . | | .     p ∈ A.pre
| . | | .     ⇒
| . | | .     p.kind = kind(a)
| . | | .     ⇒
| . | | .     [[p.rel]] rho A.ds dec_lookup(A.da;...) de e s ... ...
| . | | . 21. ∀ef:eff().
| . | | .     ef ∈ A.eff
| . | | .     ⇒
| . | | .     ef.kind = kind(a)
| . | | .     ⇒
| . | | .     x'.ef.smt.lbl
| . | | .     =
| . | | .     [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | | . 22. ∀fr:frame().
| . | | .     fr ∈ A.frame
| . | | .     ⇒
| . | | .     ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | | . 23. rel_mng_2(r;
| . | | .               rho;
```

```
| . | | .               A.ds;
| . | | .               dec_lookup(A.da;kind(a));
| . | | .               de;
| . | | .               e;
| . | | .               s;
| . | | .               x';
| . | | .               value(a);
| . | | .               tr)
| . | | . 24. r@0 : rel()
| . | | . 25. as : (Label×Term) List
| . | | . 26. 1of(unzip(as)) = rel_primed_vars(r)
| . | | . 27. ∀i:ℕ.
| . | | .     i<||as||
| . | | .     ⇒
| . | | .     2of(as[i]) ∈ ...
| . | | .                  (1of(as[i]))
| . | | . 28. r@0 = rel_subst2(as;r)
| . | | . 29. x : Label
| . | | . 30. (x ∈ 1of(unzip(as)))
| . | | . 31. i : ℕ
| . | | . 32. i<||as||
| . | | . 33. <x,apply_alist(as;x;x)> = as[i]
| . | | . 34. xxx : Term
| . | | . 35. 2of(as[i]) = xxx
| . | | . 36. xx : Label
| . | | . 37. 1of(as[i]) = xx
| . | | . 38. s1 : smt()
| . | | . 39. e1 : eff()
| . | | . 40. e1 ∈ A.eff
| . | | . 41. e1.kind =1 kind(a)
| . | | . 42. s1 = e1.smt
| . | | . 43. e1.kind = kind(a)
| . | | . 44. x'.e1.smt.lbl
| . | | .     =
| . | | .     [[e1.smt.term]] 1of(e) s value(a) niltrace()
| . | | . 45. t : SimpleType
| . | | . 46. mk_dec(e1.smt.lbl, t) ∈ A.ds
| . | | . 47. mk_dec(e1.kind, e1.typ) ∈ A.da
| . | | . 48. mk_dec(e1.smt.lbl, e1.smt.typ) ∈ A.ds
| . | | . 49. e1.smt.typ ∈ term_types(A.ds;<e1.typ>;de;e1.smt.term)
| . | | . ⊢  [[e1.smt.term]] 1of(e) s value(a) tr ∼ ...
| . | | .   by ...
| . | | .
| . | | .      ...
| . | |
| . | .\
| . | . 1. A : ioa{i:l}()
| . | . 2. r : rel()
| . | . 3. rho : Decl
| . | . 4. de : sig()
| . | . 5. e : {[[de]] rho}
| . | . 6. a : (Σ[[A.da]] rho)
| . | . 7. tr : trace_env([[A.da]] rho)
| . | . 8. tc_ioa(A;de)
| . | . 9. ¬ioa_mentions_trace(A)
```

```
| . | . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . 11. single_valued_decls(A.ds)
| . | . 12. s : [[A]] rho de e.state
| . | . 13. x' : [[A]] rho de e.state
| . | . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . 15. covers_rel(A;r)
| . | . 16. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . 17. ∀p:pre().
| . | . |   p ∈ A.pre
| . | . |   ⇒
| . | . |   p.kind = kind(a)
| . | . |   ⇒
| . | . |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | . 18. ∀ef:eff().
| . | . |   ef ∈ A.eff
| . | . |   ⇒
| . | . |   ef.kind = kind(a)
| . | . |   ⇒
| . | . |   x'.ef.smt.lbl
| . | . |   =
| . | . |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . 19. ∀fr:frame().
| . | . |   fr ∈ A.frame
| . | . |   ⇒
| . | . |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . 20. rel_mng_2(r;
| . | . |             rho;
| . | . |             A.ds;
| . | . |             dec_lookup(A.da;kind(a));
| . | . |             de;
| . | . |             e;
| . | . |             s;
| . | . |             x';
| . | . |             value(a);
| . | . |             tr)
| . | . 21. r@0 : rel()
| . | . 22. as : (Label×Term) List
| . | . 23. 1of(unzip(as)) = rel_primed_vars(r)
| . | . 24. ∀i:ℕ.
| . | . |   i<||as||
| . | . |   ⇒
| . | . |   2of(as[i]) ∈ ...
| . | . |               (1of(as[i]))
| . | . 25. r@0 = rel_subst2(as;r)
| . | . 26. x : Label
| . | . 27. (x ∈ 1of(unzip(as)))
| . | . 28. i : ℕ
| . | . 29. i<||as||
| . | . 30. <x,apply_alist(as;x;x)> = as[i]
| . | . 31. xxx : Term
| . | . 32. 2of(as[i]) = xxx
| . | . 33. xx : Label
| . | . 34. 1of(as[i]) = xx
| . | . 35. s1 : smt()
| . | . 36. f : frame()
```

```
| . | . 37. f ∈ A.frame
| . | . 38. ¬₂kind(a) ∈ f.acts
| . | . 39. s1 = mk_smt(f.var, f.var, f.typ)
| . | . ⊢  ∀t:SimpleType.
| . | .  |  mk_dec(f.var, t) ∈ A.ds
| . | .  |  ⇒
| . | .  |  t ∈ dec_lookup(A.ds;f.var) & x'.f.var = s.f.var
| . | .  | by ...
| . | . THEN AllHyps (\i. InstHyp [f] i THENA Complete Auto)
| . | .  |
| . | . THEN Auto
| . | .
| . | .  |    THEN Try (BackThruLemma `member_dec_lookup` THEN Auto)
| . | . •
| . | .  |
| . | .  |
| . | . 1. A : ioa{i:l}()
| . | . 2. r : rel()
| . | . 3. rho : Decl
| . | . 4. de : sig()
| . | . 5. e : {[[de]] rho}
| . | . 6. a : (Σ[[A.da]] rho)
| . | . 7. tr : trace_env([[A.da]] rho)
| . | . 8. tc_ioa(A;de)
| . | . 9. ¬ioa_mentions_trace(A)
| . | . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . 11. single_valued_decls(A.ds)
| . | . 12. s : [[A]] rho de e.state
| . | . 13. x' : [[A]] rho de e.state
| . | . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . 15. covers_rel(A;r)
| . | . 16. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . 17. ∀p:pre().
| . | .  |   p ∈ A.pre
| . | .  |   ⇒
| . | .  |   p.kind = kind(a)
| . | .  |   ⇒
| . | .  |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | . 18. ∀ef:eff().
| . | .  |   ef ∈ A.eff
| . | .  |   ⇒
| . | .  |   ef.kind = kind(a)
| . | .  |   ⇒
| . | .  |   x'.ef.smt.lbl
| . | .  |   =
| . | .  |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . 19. ∀fr:frame().
| . | .  |   fr ∈ A.frame
| . | .  |   ⇒
| . | .  |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . 20. rel_mng_2(r;
| . | .  |             rho;
| . | .  |             A.ds;
| . | .  |             dec_lookup(A.da;kind(a));
| . | .  |             de;
```

```
| . | . |            e;
| . | . |            s;
| . | . |            x';
| . | . |            value(a);
| . | . |            tr)
| . | . 21. r@0 : rel()
| . | . 22. as : (Label×Term) List
| . | . 23. 1of(unzip(as)) = rel_primed_vars(r)
| . | . 24. ∀i:ℕ.
| . | . |   i<||as||
| . | . |   ⇒
| . | . |   2of(as[i]) ∈ ...
| . | . |               (1of(as[i]))
| . | . 25. r@0 = rel_subst2(as;r)
| . | . 26. x : Label
| . | . 27. (x ∈ 1of(unzip(as)))
| . | . 28. i : ℕ
| . | . 29. i<||as||
| . | . 30. <x,apply_alist(as;x;x)> = as[i]
| . | . 31. xxx : Term
| . | . 32. 2of(as[i]) = xxx
| . | . 33. xx : Label
| . | . 34. 1of(as[i]) = xx
| . | . 35. s1 : smt()
| . | . 36. f : frame()
| . | . 37. f ∈ A.frame
| . | . 38. ¬(kind(a) ∈ f.acts)
| . | . 39. s1 = mk_smt(f.var, f.var, f.typ)
| . | . 40. x'.f.var = s.f.var
| . | . 41. t : SimpleType
| . | . 42. mk_dec(f.var, t) ∈ A.ds
| . | . ⊢  x'.f.var = s.f.var
| . | . | by Try (Fold `r_select` 0) THEN Subst t ∼ f.typ  0 THEN Auto
| . | . |
| . | . •
| . | . |
| . | . 1. A : ioa{i:l}()
| . | . 2. r : rel()
| . | . 3. rho : Decl
| . | . 4. de : sig()
| . | . 5. e : {[[de]] rho}
| . | . 6. a : (Σ[[A.da]] rho)
| . | . 7. tr : trace_env([[A.da]] rho)
| . | . 8. tc_ioa(A;de)
| . | . 9. ¬ioa_mentions_trace(A)
| . | . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . 11. single_valued_decls(A.ds)
| . | . 12. s : [[A]] rho de e.state
| . | . 13. x' : [[A]] rho de e.state
| . | . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . 15. covers_rel(A;r)
| . | . 16. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . 17. ∀p:pre().
| . | . |   p ∈ A.pre
| . | . |   ⇒
```

```
| . | . |   p.kind = kind(a)
| . | . |   ⇒
| . | . |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | . 18. ∀ef:eff().
| . | . |   ef ∈ A.eff
| . | . |   ⇒
| . | . |   ef.kind = kind(a)
| . | . |   ⇒
| . | . |   x'.ef.smt.lbl
| . | . |   =
| . | . |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . 19. ∀fr:frame().
| . | . |   fr ∈ A.frame
| . | . |   ⇒
| . | . |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . 20. rel_mng_2(r;
| . | . |             rho;
| . | . |             A.ds;
| . | . |             dec_lookup(A.da;kind(a));
| . | . |             de;
| . | . |             e;
| . | . |             s;
| . | . |             x';
| . | . |             value(a);
| . | . |             tr)
| . | . 21. r@0 : rel()
| . | . 22. as : (Label×Term) List
| . | . 23. 1of(unzip(as)) = rel_primed_vars(r)
| . | . 24. ∀i:ℕ.
| . | . |   i<||as||
| . | . |   ⇒
| . | . |   2of(as[i]) ∈ ...
| . | . |               (1of(as[i]))
| . | . 25. r@0 = rel_subst2(as;r)
| . | . 26. x : Label
| . | . 27. (x ∈ 1of(unzip(as)))
| . | . 28. i : ℕ
| . | . 29. i<||as||
| . | . 30. <x,apply_alist(as;x;x)> = as[i]
| . | . 31. xxx : Term
| . | . 32. 2of(as[i]) = xxx
| . | . 33. xx : Label
| . | . 34. 1of(as[i]) = xx
| . | . 35. s1 : smt()
| . | . 36. f : frame()
| . | . 37. f ∈ A.frame
| . | . 38. ¬(kind(a) ∈ f.acts)
| . | . 39. s1 = mk_smt(f.var, f.var, f.typ)
| . | . 40. x'.f.var = s.f.var
| . | . 41. t : SimpleType
| . | . 42. mk_dec(f.var, t) ∈ A.ds
| . | . ⊢  t = f.typ
| . | . | by ...f.var] i THENA Auto
| . | .
| . | . |             THEN BackThruHyp (-1) THEN Auto)·
```

```
| . | . |
| . | . 1. A : ioa{i:l}()
| . | . 2. r : rel()
| . | . 3. rho : Decl
| . | . 4. de : sig()
| . | . 5. e : {[[de]] rho}
| . | . 6. a : (Σ[[A.da]] rho)
| . | . 7. tr : trace_env([[A.da]] rho)
| . | . 8. tc_ioa(A;de)
| . | . 9. ¬ioa_mentions_trace(A)
| . | . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . 11. ∀x:Label, t1,t2:SimpleType.
| . | .     mk_dec(x, t1) ∈ A.ds ⇒ mk_dec(x, t2) ∈ A.ds ⇒ t1 = t2
| . | . 12. s : [[A]] rho de e.state
| . | . 13. x' : [[A]] rho de e.state
| . | . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . 15. covers_rel(A;r)
| . | . 16. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . 17. ∀p:pre().
| . | .     p ∈ A.pre
| . | .     ⇒
| . | .     p.kind = kind(a)
| . | .     ⇒
| . | .     [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | . 18. ∀ef:eff().
| . | .     ef ∈ A.eff
| . | .     ⇒
| . | .     ef.kind = kind(a)
| . | .     ⇒
| . | .     x'.ef.smt.lbl
| . | .     =
| . | .     [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . 19. ∀fr:frame().
| . | .     fr ∈ A.frame
| . | .     ⇒
| . | .     ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . 20. rel_mng_2(r;
| . | .               rho;
| . | .               A.ds;
| . | .               dec_lookup(A.da;kind(a));
| . | .               de;
| . | .               e;
| . | .               s;
| . | .               x';
| . | .               value(a);
| . | .               tr)
| . | . 21. r@0 : rel()
| . | . 22. as : (Label×Term) List
| . | . 23. 1of(unzip(as)) = rel_primed_vars(r)
| . | . 24. ∀i:ℕ.
| . | .     i<||as||
| . | .     ⇒
| . | .     2of(as[i]) ∈ ...
| . | .                  (1of(as[i]))
| . | . 25. r@0 = rel_subst2(as;r)
```

```
| . | . 26. x : Label
| . | . 27. (x ∈ 1of(unzip(as)))
| . | . 28. i : ℕ
| . | . 29. i<||as||
| . | . 30. <x,apply_alist(as;x;x)> = as[i]
| . | . 31. xxx : Term
| . | . 32. 2of(as[i]) = xxx
| . | . 33. xx : Label
| . | . 34. 1of(as[i]) = xx
| . | . 35. s1 : smt()
| . | . 36. f : frame()
| . | . 37. f ∈ A.frame
| . | . 38. ¬(kind(a) ∈ f.acts)
| . | . 39. s1 = mk_smt(f.var, f.var, f.typ)
| . | . 40. x'.f.var = s.f.var
| . | . 41. t : SimpleType
| . | . 42. mk_dec(f.var, t) ∈ A.ds
| . | . 43. ∀t1,t2:SimpleType.
| . | .     mk_dec(f.var, t1) ∈ A.ds
| . | .     ⇒
| . | .     mk_dec(f.var, t2) ∈ A.ds ⇒ t1 = t2
| . | . ⊢  mk_dec(f.var, f.typ) ∈ A.ds
| . | .   by ...•
| . |
| . .\
| . . 1. A : ioa{i:l}()
| . . 2. r : rel()
| . . 3. rho : Decl
| . . 4. de : sig()
| . . 5. e : {[[de]] rho}
| . . 6. a : (Σ[[A.da]] rho)
| . . 7. tr : trace_env([[A.da]] rho)
| . . 8. tc_ioa(A;de)
| . . 9. ¬ioa_mentions_trace(A)
| . . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . . 11. single_valued_decls(A.ds)
| . . 12. s : [[A]] rho de e.state
| . . 13. x' : [[A]] rho de e.state
| . . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . . 15. covers_rel(A;r)
| . . 16. ∀p:pre().
| . .     p ∈ A.pre
| . .     ⇒
| . .     p.kind = kind(a)
| . .     ⇒
| . .     [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . . 17. ∀ef:eff().
| . .     ef ∈ A.eff
| . .     ⇒
| . .     ef.kind = kind(a)
| . .     ⇒
| . .     x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . . 18. ∀fr:frame().
| . .     fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . . 19. rel_mng_2(r;
```

```
| . .                rho;
| . .                A.ds;
| . .                dec_lookup(A.da;kind(a));
| . .                de;
| . .                e;
| . .                s;
| . .                x';
| . .                value(a);
| . .                tr)
| . . 20. r@0 : rel()
| . . 21. as : (Label×Term) List
| . . 22. 1of(unzip(as)) = rel_primed_vars(r)
| . . 23. ∀i:ℕ.
| . .     i<||as||
| . .     ⇒
| . .     2of(as[i]) ∈ (λx.
| . .                    smts_eff(action_effect(kind(a);A.eff;...);x))
| . .                   (1of(as[i]))
| . . 24. r@0 = rel_subst2(as;r)
| . . 25. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . . 26. tc(rel_subst2(as;r);A.ds;dec_lookup(A.da;kind(a));de)
| . .     & trace_consistent_rel(rho;A.da;tr.proj;rel_subst2(as;r))
| . .     & ([[rel_subst2(as;r)]] rho A.ds ... de e s value(a) tr
| . .        ⟺
| . .        rel_mng_2(r;
| . .                  rho;
| . .                  A.ds;
| . .                  dec_lookup(A.da;kind(a));
| . .                  de;
| . .                  e;
| . .                  s;
| . .                  x';
| . .                  value(a);
| . .                  tr))
| . . ⊢  [[rel_subst2(as;r)]] rho A.ds ... de e s value(a) tr
| . .   by ExRepD THEN D (-1) THEN D (-1) THEN Try Trivial
| . .  •
|
|\
| 1. A : ioa{i:l}()
| 2. r : rel()
| 3. rho : Decl
| 4. de : sig()
| 5. e : {[[de]] rho}
| 6. a : (Σ[[A.da]] rho)
| 7. tr : trace_env([[A.da]] rho)
| 8. tc_ioa(A;de)
| 9. ¬ioa_mentions_trace(A)
| 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| 11. single_valued_decls(A.ds)
| 12. s : [[A]] rho de e.state
| 13. x' : [[A]] rho de e.state
| 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| 15. covers_rel(A;r)
| 16. [[A]] rho de e.trans(s,a,x')
```

```
| 17. [[wp2_rel(A;kind(a);r)]] rho A.ds ... de e s value(a) tr
| ⊢  rel_mng_2(r;
| |            rho;
| |            A.ds;
| |            dec_lookup(A.da;kind(a));
| |            de;
| |            e;
| |            s;
| |            x';
| |            value(a);
| |            tr)
| | by Unfold `ioa_mng` (-2) THEN Reduce (-2) THEN RepD
| |
| 1. A : ioa{i:l}()
| 2. r : rel()
| 3. rho : Decl
| 4. de : sig()
| 5. e : {[[de]] rho}
| 6. a : (Σ[[A.da]] rho)
| 7. tr : trace_env([[A.da]] rho)
| 8. tc_ioa(A;de)
| 9. ¬ioa_mentions_trace(A)
| 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| 11. single_valued_decls(A.ds)
| 12. s : [[A]] rho de e.state
| 13. x' : [[A]] rho de e.state
| 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| 15. covers_rel(A;r)
| 16. ∀p:pre().
| |   p ∈ A.pre
| |   ⇒
| |   p.kind = kind(a)
| |   ⇒
| |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s value(a) ...
| 17. ∀ef:eff().
| |   ef ∈ A.eff
| |   ⇒
| |   ef.kind = kind(a)
| |   ⇒
| |   x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| 18. ∀fr:frame().
| |   fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| 19. [[wp2_rel(A;kind(a);r)]] rho A.ds ... de e s value(a) tr
| ⊢  rel_mng_2(r;
| |            rho;
| |            A.ds;
| |            dec_lookup(A.da;kind(a));
| |            de;
| |            e;
| |            s;
| |            x';
| |            value(a);
| |            tr)
| | by Assert ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| |
```

```
| |\
| | .....assertion .....
| | 1. A : ioa{i:l}()
| | 2. r : rel()
| | 3. rho : Decl
| | 4. de : sig()
| | 5. e : {[[de]] rho}
| | 6. a : (Σ[[A.da]] rho)
| | 7. tr : trace_env([[A.da]] rho)
| | 8. tc_ioa(A;de)
| | 9. ¬ioa_mentions_trace(A)
| | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| | 11. single_valued_decls(A.ds)
| | 12. s : [[A]] rho de e.state
| | 13. x' : [[A]] rho de e.state
| | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| | 15. covers_rel(A;r)
| | 16. ∀p:pre().
| |     p ∈ A.pre
| |     ⇒
| |     p.kind = kind(a)
| |     ⇒
| |     [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| | 17. ∀ef:eff().
| |     ef ∈ A.eff
| |     ⇒
| |     ef.kind = kind(a)
| |     ⇒
| |     x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| | 18. ∀fr:frame().
| |     fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| | 19. [[wp2_rel(A;kind(a);r)]] rho A.ds ... de e s value(a) tr
| | ⊢  ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| |   by UnivCD THENA Auto THEN
| |
| |      All (\i.Unfold `covers_rel` i THEN BackThruHyp i )
| |
| |      THENM AutoInstConcl []
| | THENA Auto
| | THEN Fold `pred` 0

| |
| |      THEN Auto•
| |
| .\
| . 1. A : ioa{i:l}()
| . 2. r : rel()
| . 3. rho : Decl
| . 4. de : sig()
| . 5. e : {[[de]] rho}
| . 6. a : (Σ[[A.da]] rho)
| . 7. tr : trace_env([[A.da]] rho)
| . 8. tc_ioa(A;de)
| . 9. ¬ioa_mentions_trace(A)
| . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . 11. single_valued_decls(A.ds)
```

```
| . 12. s : [[A]] rho de e.state
| . 13. x' : [[A]] rho de e.state
| . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . 15. covers_rel(A;r)
| . 16. ∀p:pre().
| . |   p ∈ A.pre
| . |   ⇒
| . |   p.kind = kind(a)
| . |   ⇒
| . |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . 17. ∀ef:eff().
| . |   ef ∈ A.eff
| . |   ⇒
| . |   ef.kind = kind(a)
| . |   ⇒
| . |   x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . 18. ∀fr:frame().
| . |   fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . 19. [[wp2_rel(A;kind(a);r)]] rho A.ds ... de e s value(a) tr
| . 20. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . ⊢  rel_mng_2(r;
| . |            rho;
| . |            A.ds;
| . |            dec_lookup(A.da;kind(a));
| . |            de;
| . |            e;
| . |            s;
| . |            x';
| . |            value(a);
| . |            tr)
| . | by Assert value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho THEN
| .
| . |    Try (AllHyps (Unfold `ioa_mng`) THEN AllHyps Reduce
| .
| . |    ...
| . |
| . 1. A : ioa{i:l}()
| . 2. r : rel()
| . 3. rho : Decl
| . 4. de : sig()
| . 5. e : {[[de]] rho}
| . 6. a : (Σ[[A.da]] rho)
| . 7. tr : trace_env([[A.da]] rho)
| . 8. tc_ioa(A;de)
| . 9. ¬ioa_mentions_trace(A)
| . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . 11. single_valued_decls(A.ds)
| . 12. s : [[A]] rho de e.state
| . 13. x' : [[A]] rho de e.state
| . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . 15. covers_rel(A;r)
| . 16. ∀p:pre().
| . |   p ∈ A.pre
| . |   ⇒
| . |   p.kind = kind(a)
```

```
| . |   ⇒
| . |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . 17. ∀ef:eff().
| . |   ef ∈ A.eff
| . |   ⇒
| . |   ef.kind = kind(a)
| . |   ⇒
| . |   x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . 18. ∀fr:frame().
| . |   fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . 19. [[wp2_rel(A;kind(a);r)]] rho A.ds ... de e s value(a) tr
| . 20. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . 21. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . ⊢  rel_mng_2(r;
| . |            rho;
| . |            A.ds;
| . |            dec_lookup(A.da;kind(a));
| . |            de;
| . |            e;
| . |            s;
| . |            x';
| . |            value(a);
| . |            tr)
| . | by MoveToHyp 0 (-3) THEN Unfold `wp2_rel` (-1)
| .
| . |      THEN Unfold `pred_mng` (-1)
| . THENA Auto•
| . |
| . 1. A : ioa{i:l}()
| . 2. r : rel()
| . 3. rho : Decl
| . 4. de : sig()
| . 5. e : {[[de]] rho}
| . 6. a : (Σ[[A.da]] rho)
| . 7. tr : trace_env([[A.da]] rho)
| . 8. tc_ioa(A;de)
| . 9. ¬ioa_mentions_trace(A)
| . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . 11. single_valued_decls(A.ds)
| . 12. s : [[A]] rho de e.state
| . 13. x' : [[A]] rho de e.state
| . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . 15. covers_rel(A;r)
| . 16. ∀p:pre().
| . |   p ∈ A.pre
| . |   ⇒
| . |   p.kind = kind(a)
| . |   ⇒
| . |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . 17. ∀ef:eff().
| . |   ef ∈ A.eff
| . |   ⇒
| . |   ef.kind = kind(a)
| . |   ⇒
| . |   x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
```

```
| . 18. ∀fr:frame().
| . |   fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . 19. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . 20. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . 21. ∀r@0:rel().
| . |   r@0 ∈ col_subst2(λx.
| . |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . |   ⇒
| . |   [[r@0]] rho A.ds dec_lookup(A.da;kind(a)) de e s value(a) tr
| . ⊢  rel_mng_2(r;
| . |            rho;
| . |            A.ds;
| . |            dec_lookup(A.da;kind(a));
| . |            de;
| . |            e;
| . |            s;
| . |            x';
| . |            value(a);
| . |            tr)
| . | by Assert ∃r':rel().
| . |    r' ∈ col_subst2(λx.
| . |    smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . |    & tc(r';A.ds;dec_lookup(A.da;kind(a));de)
| . |      & trace_consistent_rel(rho;A.da;tr.proj;r')
| . |      & ([[r']] rho A.ds dec_lookup(A.da;kind(a)) de e s ... tr
| . |         ⇒
| . |         rel_mng_2(r;
| . |                   rho;
| . |                   A.ds;
| . |                   dec_lookup(A.da;kind(a));
| . |                   de;
| . |                   e;
| . |                   s;
| . |                   x';
| . |                   value(a);
| . |                   tr))•
| . |
| . |\
| . | .....assertion .....
| . | 1. A : ioa{i:l}()
| . | 2. r : rel()
| . | 3. rho : Decl
| . | 4. de : sig()
| . | 5. e : {[[de]] rho}
| . | 6. a : (Σ[[A.da]] rho)
| . | 7. tr : trace_env([[A.da]] rho)
| . | 8. tc_ioa(A;de)
| . | 9. ¬ioa_mentions_trace(A)
| . | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | 11. single_valued_decls(A.ds)
| . | 12. s : [[A]] rho de e.state
| . | 13. x' : [[A]] rho de e.state
| . | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | 15. covers_rel(A;r)
| . | 16. ∀p:pre().
```

```
| . | |   p ∈ A.pre
| . | |   ⇒
| . | |   p.kind = kind(a)
| . | |   ⇒
| . | |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | 17. ∀ef:eff().
| . | |   ef ∈ A.eff
| . | |   ⇒
| . | |   ef.kind = kind(a)
| . | |   ⇒
| . | |   x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | 18. ∀fr:frame().
| . | |   fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | 19. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | 20. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | 21. ∀r@0:rel().
| . | |   r@0 ∈ col_subst2(λx.
| . | |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | |   ⇒
| . | |   [[r@0]] rho A.ds dec_lookup(A.da;kind(a)) de e s value(a) tr
| . | ⊢  ∃r':rel().
| . | |  r' ∈ col_subst2(λx.
| . | |  smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | |  & tc(r';A.ds;dec_lookup(A.da;kind(a));de)
| . | |    & trace_consistent_rel(rho;A.da;tr.proj;r')
| . | |    & ([[r']] rho A.ds dec_lookup(A.da;kind(a)) de e s ... tr
| . | |       ⇒
| . | |       rel_mng_2(r;
| . | |                 rho;
| . | |                 A.ds;
| . | |                 dec_lookup(A.da;kind(a));
| . | |                 de;
| . | |                 e;
| . | |                 s;
| . | |                 x';
| . | |                 value(a);
| . | |                 tr))
| . | | by ...
| . | THEN Assert ∃r':rel().
| . | |    r' ∈ col_subst2(λx.
| . | |    smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | |
| . | |\
| . | | .....assertion .....
| . | | 1. A : ioa{i:l}()
| . | | 2. r : rel()
| . | | 3. rho : Decl
| . | | 4. de : sig()
| . | | 5. e : {[[de]] rho}
| . | | 6. a : (Σ[[A.da]] rho)
| . | | 7. tr : trace_env([[A.da]] rho)
| . | | 8. tc_ioa(A;de)
| . | | 9. ¬ioa_mentions_trace(A)
| . | | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | | 11. single_valued_decls(A.ds)
```

```
| . | | 12. s : {[[A.ds]] rho}
| . | | 13. x' : {[[A.ds]] rho}
| . | | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | | 15. covers_rel(A;r)
| . | | 16. ∀p:pre().
| . | | |   p ∈ A.pre
| . | | |   ⇒
| . | | |   p.kind = kind(a)
| . | | |   ⇒
| . | | |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | | 17. ∀ef:eff().
| . | | |   ef ∈ A.eff
| . | | |   ⇒
| . | | |   ef.kind = kind(a)
| . | | |   ⇒
| . | | |   x'.ef.smt.lbl
| . | | |   =
| . | | |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | | 18. ∀fr:frame().
| . | | |   fr ∈ A.frame
| . | | |   ⇒
| . | | |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | | 19. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | | 20. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | | 21. ∀r@0:rel().
| . | | |   r@0 ∈ col_subst2(λx.
| . | | |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | | |   ⇒
| . | | |   [[r@0]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... tr
| . | | ⊢  ∃r':rel().
| . | | |  r' ∈ col_subst2(λx.
| . | | |  smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | | | by InstLemma `covers_pred_lemma2` [r;r;A;kind(a)] THEN Auto
| . | | |
| . | | THEN Unfold `pred_rel` 0
| . | |
| . | | |    THEN Try (Unfold `covers_pred` 0 )
| . | |
| . | | |    THEN Try (RW ColMemberC 0 THEN Auto)
| . | | •
| . | | |
| . | | 1. A : ioa{i:l}()
| . | | 2. r : rel()
| . | | 3. rho : Decl
| . | | 4. de : sig()
| . | | 5. e : {[[de]] rho}
| . | | 6. a : (Σ[[A.da]] rho)
| . | | 7. tr : trace_env([[A.da]] rho)
| . | | 8. tc_ioa(A;de)
| . | | 9. ¬ioa_mentions_trace(A)
| . | | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | | 11. single_valued_decls(A.ds)
| . | | 12. s : {[[A.ds]] rho}
| . | | 13. x' : {[[A.ds]] rho}
| . | | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
```

```
| . | | 15. covers_rel(A;r)
| . | | 16. ∀p:pre().
| . | | |   p ∈ A.pre
| . | | |   ⇒
| . | | |   p.kind = kind(a)
| . | | |   ⇒
| . | | |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | | 17. ∀ef:eff().
| . | | |   ef ∈ A.eff
| . | | |   ⇒
| . | | |   ef.kind = kind(a)
| . | | |   ⇒
| . | | |   x'.ef.smt.lbl
| . | | |   =
| . | | |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | | 18. ∀fr:frame().
| . | | |   fr ∈ A.frame
| . | | |   ⇒
| . | | |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | | 19. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | | 20. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | | 21. ∀r@0:rel().
| . | | |   r@0 ∈ col_subst2(λx.
| . | | |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | | |   ⇒
| . | | |   [[r@0]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... tr
| . | | ⊢  ∀x:Label. pred_mentions(<r>;x) ⇒ covers_var(A;x)
| . | | | by AllHyps (\h. Unfold `covers_rel` h THEN ParallelOp h
| . | |
| . | | |                        THEN ParallelOp (-1))
| . | | |
| . | | .....antecedent .....
| . | | 1. A : ioa{i:l}()
| . | | 2. r : rel()
| . | | 3. rho : Decl
| . | | 4. de : sig()
| . | | 5. e : {[[de]] rho}
| . | | 6. a : (Σ[[A.da]] rho)
| . | | 7. tr : trace_env([[A.da]] rho)
| . | | 8. tc_ioa(A;de)
| . | | 9. ¬ioa_mentions_trace(A)
| . | | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | | 11. single_valued_decls(A.ds)
| . | | 12. s : {[[A.ds]] rho}
| . | | 13. x' : {[[A.ds]] rho}
| . | | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | | 15. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | | 16. ∀p:pre().
| . | |     p ∈ A.pre
| . | |     ⇒
| . | |     p.kind = kind(a)
| . | |     ⇒
| . | |     [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | | 17. ∀ef:eff().
| . | |     ef ∈ A.eff
```

```
| . | |     ⇒
| . | |     ef.kind = kind(a)
| . | |     ⇒
| . | |     x'.ef.smt.lbl
| . | |     =
| . | |     [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | | 18. ∀fr:frame().
| . | |     fr ∈ A.frame
| . | |     ⇒
| . | |     ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | | 19. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | | 20. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | | 21. ∀r@0:rel().
| . | |     r@0 ∈ col_subst2(λx.
| . | |     smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | |     ⇒
| . | |     [[r@0]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... tr
| . | | 22. x : Label
| . | | 23. pred_mentions(<r>;x)
| . | | ⊢  rel_mentions(r;x)
| . | |   by ...
| . | | THEN HypSubstSq (-2) (-1) THEN Auto
| . | |
| . | .\
| . | . 1. A : ioa{i:l}()
| . | . 2. r : rel()
| . | . 3. rho : Decl
| . | . 4. de : sig()
| . | . 5. e : {[[de]] rho}
| . | . 6. a : (Σ[[A.da]] rho)
| . | . 7. tr : trace_env([[A.da]] rho)
| . | . 8. tc_ioa(A;de)
| . | . 9. ¬ioa_mentions_trace(A)
| . | . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . 11. single_valued_decls(A.ds)
| . | . 12. s : {[[A.ds]] rho}
| . | . 13. x' : {[[A.ds]] rho}
| . | . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . 15. covers_rel(A;r)
| . | . 16. ∀p:pre().
| . | .   |   p ∈ A.pre
| . | .   |   ⇒
| . | .   |   p.kind = kind(a)
| . | .   |   ⇒
| . | .   |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . | . 17. ∀ef:eff().
| . | .   |   ef ∈ A.eff
| . | .   |   ⇒
| . | .   |   ef.kind = kind(a)
| . | .   |   ⇒
| . | .   |   x'.ef.smt.lbl
| . | .   |   =
| . | .   |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . 18. ∀fr:frame().
| . | .   |   fr ∈ A.frame
```

```
| . | . |   ⇒
| . | . |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . 19. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . 20. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . 21. ∀r@0:rel().
| . | . |   r@0 ∈ col_subst2(λx.
| . | . |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . |   ⇒
| . | . |   [[r@0]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... tr
| . | . 22. ∃r':rel().
| . | . |   r' ∈ col_subst2(λx.
| . | . |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . ⊢  ∃r':rel().
| . | . |  r' ∈ col_subst2(λx.
| . | . |  smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . |  & tc(r';A.ds;dec_lookup(A.da;kind(a));de)
| . | . |    & trace_consistent_rel(rho;A.da;tr.proj;r')
| . | . |    & ([[r']] rho A.ds dec_lookup(A.da;kind(a)) de e s ... tr
| . | . |       ⇒
| . | . |       rel_mng_2(r;
| . | . |                 rho;
| . | . |                 A.ds;
| . | . |                 dec_lookup(A.da;kind(a));
| . | . |                 de;
| . | . |                 e;
| . | . |                 s;
| . | . |                 x';
| . | . |                 value(a);
| . | . |                 tr))
| . | . | by Assert tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . |
| . | . |\
| . | . | .....assertion .....
| . | . | 1. A : ioa{i:l}()
| . | . | 2. r : rel()
| . | . | 3. rho : Decl
| . | . | 4. de : sig()
| . | . | 5. e : {[[de]] rho}
| . | . | 6. a : (Σ[[A.da]] rho)
| . | . | 7. tr : trace_env([[A.da]] rho)
| . | . | 8. tc_ioa(A;de)
| . | . | 9. ¬ioa_mentions_trace(A)
| . | . | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . | 11. single_valued_decls(A.ds)
| . | . | 12. s : {[[A.ds]] rho}
| . | . | 13. x' : {[[A.ds]] rho}
| . | . | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . | 15. covers_rel(A;r)
| . | . | 16. ∀p:pre().
| . | . |     p ∈ A.pre
| . | . |     ⇒
| . | . |     p.kind = kind(a)
| . | . |     ⇒
| . | . |     [[p.rel]] rho A.ds dec_lookup(A.da;...) de e s ... ...
| . | . | 17. ∀ef:eff().
```

```
| . | . |     ef ∈ A.eff
| . | . |     ⇒
| . | . |     ef.kind = kind(a)
| . | . |     ⇒
| . | . |     x'.ef.smt.lbl
| . | . |     =
| . | . |     [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . | 18. ∀fr:frame().
| . | . |     fr ∈ A.frame
| . | . |     ⇒
| . | . |     ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . | 19. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . | 20. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . | 21. ∀r@0:rel().
| . | . |     r@0 ∈ col_subst2(λx.
| . | . |     smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . |     ⇒
| . | . |     [[r@0]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... tr
| . | . | 22. ∃r':rel().
| . | . |     r' ∈ col_subst2(λx.
| . | . |     smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . | ⊢  tc(r;A.ds;dec_lookup(A.da;kind(a));de)  by Trivial•
| . | . |
| . | . .\
| . | . . 1. A : ioa{i:l}()
| . | . . 2. r : rel()
| . | . . 3. rho : Decl
| . | . . 4. de : sig()
| . | . . 5. e : {[[de]] rho}
| . | . . 6. a : (Σ[[A.da]] rho)
| . | . . 7. tr : trace_env([[A.da]] rho)
| . | . . 8. tc_ioa(A;de)
| . | . . 9. ¬ioa_mentions_trace(A)
| . | . . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . 11. single_valued_decls(A.ds)
| . | . . 12. s : {[[A.ds]] rho}
| . | . . 13. x' : {[[A.ds]] rho}
| . | . . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . 15. covers_rel(A;r)
| . | . . 16. ∀p:pre().
| . | . . |   p ∈ A.pre
| . | . . |   ⇒
| . | . . |   p.kind = kind(a)
| . | . . |   ⇒
| . | . . |   [[p.rel]] rho A.ds dec_lookup(A.da;...) de e s ... ...
| . | . . 17. ∀ef:eff().
| . | . . |   ef ∈ A.eff
| . | . . |   ⇒
| . | . . |   ef.kind = kind(a)
| . | . . |   ⇒
| . | . . |   x'.ef.smt.lbl
| . | . . |   =
| . | . . |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . 18. ∀fr:frame().
| . | . . |   fr ∈ A.frame
```

```
| . | . . |   ⇒
| . | . . |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . 19. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . 20. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . 21. ∀r@0:rel().
| . | . . |   r@0 ∈ col_subst2(λx.
| . | . . |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . . |   ⇒
| . | . . |   [[r@0]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... tr
| . | . . 22. ∃r':rel().
| . | . . |   r' ∈ col_subst2(λx.
| . | . . |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . . 23. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . ⊢  ∃r':rel().
| . | . . |  r' ∈ col_subst2(λx.
| . | . . |  smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . . |  & tc(r';A.ds;dec_lookup(A.da;kind(a));de)
| . | . . |    & trace_consistent_rel(rho;A.da;tr.proj;r')
| . | . . |    & ([[r']] rho A.ds dec_lookup(A.da;...) de e s ... tr
| . | . . |       ⇒
| . | . . |       rel_mng_2(r;
| . | . . |                 rho;
| . | . . |                 A.ds;
| . | . . |                 dec_lookup(A.da;kind(a));
| . | . . |                 de;
| . | . . |                 e;
| . | . . |                 s;
| . | . . |                 x';
| . | . . |                 value(a);
| . | . . |                 tr))
| . | . . | by ExRepD THEN InstConcl [r'] THENM D 0
| . | . . THENM Try Trivial
| . | . . |
| . | . . THENA Auto
| . | . . THEN Try (Reduce 0 THEN Complete Auto)
| . | . . |
| . | . . |    .
| . | . . |
| . | . . |\
| . | . . | 1. A : ioa{i:l}()
| . | . . | 2. r : rel()
| . | . . | 3. rho : Decl
| . | . . | 4. de : sig()
| . | . . | 5. e : {[[de]] rho}
| . | . . | 6. a : (Σ[[A.da]] rho)
| . | . . | 7. tr : trace_env([[A.da]] rho)
| . | . . | 8. tc_ioa(A;de)
| . | . . | 9. ¬ioa_mentions_trace(A)
| . | . . | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | 11. single_valued_decls(A.ds)
| . | . . | 12. s : {[[A.ds]] rho}
| . | . . | 13. x' : {[[A.ds]] rho}
| . | . . | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | 15. covers_rel(A;r)
| . | . . | 16. ∀p:pre().
```

```
| . | . . | |   p ∈ A.pre
| . | . . | |   ⇒
| . | . . | |   p.kind = kind(a)
| . | . . | |   ⇒
| . | . . | |   [[p.rel]] rho A.ds dec_lookup(A.da;...) de e s ... ...
| . | . . | 17. ∀ef:eff().
| . | . . | |   ef ∈ A.eff
| . | . . | |   ⇒
| . | . . | |   ef.kind = kind(a)
| . | . . | |   ⇒
| . | . . | |   x'.ef.smt.lbl
| . | . . | |   =
| . | . . | |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | 18. ∀fr:frame().
| . | . . | |   fr ∈ A.frame
| . | . . | |   ⇒
| . | . . | |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | 19. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | 20. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | 21. ∀r@0:rel().
| . | . . | |   r@0 ∈ col_subst2(λx.
| . | . . | |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . . | |   ⇒
| . | . . | |   [[r@0]] rho A.ds dec_lookup(A.da;...) de e s ... tr
| . | . . | 22. r' : rel()
| . | . . | 23. r' ∈ col_subst2(λx.
| . | . . | |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . . | 24. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | ⊢  tc(r';A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | |  & trace_consistent_rel(rho;A.da;tr.proj;r')
| . | . . | |  & ([[r']] rho A.ds dec_lookup(A.da;...) de e s ... tr
| . | . . | |     ⇒
| . | . . | |     rel_mng_2(r;
| . | . . | |               rho;
| . | . . | |               A.ds;
| . | . . | |               dec_lookup(A.da;kind(a));
| . | . . | |               de;
| . | . . | |               e;
| . | . . | |               s;
| . | . . | |               x';
| . | . . | |               value(a);
| . | . . | |               tr))
| . | . . | | by RWW "member_col_subst2" (-2)
| . | . . | THENM Reduce (-2)
| . | . . |
| . | . . | |    THENM ExRepD
| . | . . | THENM HypSubstSq (-2) 0
| . | . . |
| . | . . | |    THENA (Auto THEN Reduce 0 THEN Auto)
| . | . . | |
| . | . . | 1. A : ioa{i:l}()
| . | . . | 2. r : rel()
| . | . . | 3. rho : Decl
| . | . . | 4. de : sig()
| . | . . | 5. e : {[[de]] rho}
```

```
| . | . . | 6. a : (Σ[[A.da]] rho)
| . | . . | 7. tr : trace_env([[A.da]] rho)
| . | . . | 8. tc_ioa(A;de)
| . | . . | 9. ¬ioa_mentions_trace(A)
| . | . . | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | 11. single_valued_decls(A.ds)
| . | . . | 12. s : {[[A.ds]] rho}
| . | . . | 13. x' : {[[A.ds]] rho}
| . | . . | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | 15. covers_rel(A;r)
| . | . . | 16. ∀p:pre().
| . | . . | |   p ∈ A.pre
| . | . . | |   ⇒
| . | . . | |   p.kind = kind(a)
| . | . . | |   ⇒
| . | . . | |   [[p.rel]] rho A.ds dec_lookup(A.da;...) de e s ... ...
| . | . . | 17. ∀ef:eff().
| . | . . | |   ef ∈ A.eff
| . | . . | |   ⇒
| . | . . | |   ef.kind = kind(a)
| . | . . | |   ⇒
| . | . . | |   x'.ef.smt.lbl
| . | . . | |   =
| . | . . | |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | 18. ∀fr:frame().
| . | . . | |   fr ∈ A.frame
| . | . . | |   ⇒
| . | . . | |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | 19. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | 20. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | 21. ∀r@0:rel().
| . | . . | |   r@0 ∈ col_subst2(λx.
| . | . . | |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . . | |   ⇒
| . | . . | |   [[r@0]] rho A.ds dec_lookup(A.da;...) de e s ... tr
| . | . . | 22. r' : rel()
| . | . . | 23. as : (Label×Term) List
| . | . . | 24. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | 25. ∀i:ℕ.
| . | . . | |   i<||as||
| . | . . | |   ⇒
| . | . . | |   2of(as[i]) ∈ smts_eff(action_effect(...;...;...);...)
| . | . . | 26. r' = rel_subst2(as;r)
| . | . . | 27. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | ⊢  tc(rel_subst2(as;r);A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | |  & trace_consistent_rel(rho;A.da;tr.proj;...)
| . | . . | |  & ([[rel_subst2(as;r)]] rho A.ds ... de e s value(a) tr
| . | . . | |     ⇒
| . | . . | |     rel_mng_2(r;
| . | . . | |               rho;
| . | . . | |               A.ds;
| . | . . | |               dec_lookup(A.da;kind(a));
| . | . . | |               de;
| . | . . | |               e;
| . | . . | |               s;
```

```
| . | . . | |              x';
| . | . . | |              value(a);
| . | . . | |              tr))
| . | . . | | by InstLemma `rel_subst_mng_2_iff` [r;as;A.ds;A.da;
| . | . . | |    dec_lookup(A.da;kind(a));de;rho;e;s;x';kind(a);
| . | . . | |    value(a);tr] THENA Try (Complete Auto)
| . | . . |
| . | . . | |    THEN TrivialIoaHyp•
| . | . . | |
| . | . . | |\
| . | . . | | 1. A : ioa{i:l}()
| . | . . | | 2. r : rel()
| . | . . | | 3. rho : Decl
| . | . . | | 4. de : sig()
| . | . . | | 5. e : {[[de]] rho}
| . | . . | | 6. a : (Σ[[A.da]] rho)
| . | . . | | 7. tr : trace_env([[A.da]] rho)
| . | . . | | 8. tc_ioa(A;de)
| . | . . | | 9. ¬ioa_mentions_trace(A)
| . | . . | | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | | 11. single_valued_decls(A.ds)
| . | . . | | 12. s : {[[A.ds]] rho}
| . | . . | | 13. x' : {[[A.ds]] rho}
| . | . . | | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | 15. covers_rel(A;r)
| . | . . | | 16. ∀p:pre().
| . | . . | | |   p ∈ A.pre
| . | . . | | |   ⇒
| . | . . | | |   p.kind = kind(a)
| . | . . | | |   ⇒
| . | . . | | |   [[p.rel]] rho A.ds ... de e s value(a) niltrace()
| . | . . | | 17. ∀ef:eff().
| . | . . | | |   ef ∈ A.eff
| . | . . | | |   ⇒
| . | . . | | |   ef.kind = kind(a)
| . | . . | | |   ⇒
| . | . . | | |   x'.ef.smt.lbl
| . | . . | | |   =
| . | . . | | |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | 18. ∀fr:frame().
| . | . . | | |   fr ∈ A.frame
| . | . . | | |   ⇒
| . | . . | | |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | | 19. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | | 20. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | | 21. ∀r@0:rel().
| . | . . | | |   r@0 ∈ col_subst2(λx.
| . | . . | | |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . . | | |   ⇒
| . | . . | | |   [[r@0]] rho A.ds dec_lookup(A.da;...) de e s ... tr
| . | . . | | 22. r' : rel()
| . | . . | | 23. as : (Label×Term) List
| . | . . | | 24. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | | 25. ∀i:ℕ.
| . | . . | | |   i<||as||
```

```
| . | . . | | |   ⇒
| . | . . | | |   2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | | 26. r' = rel_subst2(as;r)
| . | . . | | 27. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | ⊢  ¬subst_mentions_trace(as)
| . | . . | | | by ...
| . | . . | | THEN RepeatFor 2 (OrLeft THENA Auto)
| . | . . | | |
| . | . . | | 1. A : ioa{i:l}()
| . | . . | | 2. r : rel()
| . | . . | | 3. rho : Decl
| . | . . | | 4. de : sig()
| . | . . | | 5. e : {[[de]] rho}
| . | . . | | 6. a : (Σ[[A.da]] rho)
| . | . . | | 7. tr : trace_env([[A.da]] rho)
| . | . . | | 8. tc_ioa(A;de)
| . | . . | | 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | | 10. single_valued_decls(A.ds)
| . | . . | | 11. s : {[[A.ds]] rho}
| . | . . | | 12. x' : {[[A.ds]] rho}
| . | . . | | 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | 14. covers_rel(A;r)
| . | . . | | 15. ∀p:pre().
| . | . . | | |   p ∈ A.pre
| . | . . | | |   ⇒
| . | . . | | |   p.kind = kind(a)
| . | . . | | |   ⇒
| . | . . | | |   [[p.rel]] rho A.ds ... de e s value(a) niltrace()
| . | . . | | 16. ∀ef:eff().
| . | . . | | |   ef ∈ A.eff
| . | . . | | |   ⇒
| . | . . | | |   ef.kind = kind(a)
| . | . . | | |   ⇒
| . | . . | | |   x'.ef.smt.lbl
| . | . . | | |   =
| . | . . | | |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | 17. ∀fr:frame().
| . | . . | | |   fr ∈ A.frame
| . | . . | | |   ⇒
| . | . . | | |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | | 18. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | | 19. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | | 20. ∀r@0:rel().
| . | . . | | |   r@0 ∈ col_subst2(λx.
| . | . . | | |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . . | | |   ⇒
| . | . . | | |   [[r@0]] rho A.ds dec_lookup(A.da;...) de e s ... tr
| . | . . | | 21. r' : rel()
| . | . . | | 22. as : (Label×Term) List
| . | . . | | 23. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | | 24. ∀i:ℕ.
| . | . . | | |   i<||as||
| . | . . | | |   ⇒
| . | . . | | |   2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | | 25. r' = rel_subst2(as;r)
```

```
| . | . . | | 26. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | 27. subst_mentions_trace(as)
| . | . . | | ⊢  ∃e:eff(). e ∈ A.eff & mentions_trace(e.smt.term)
| . | . . | | | by ...
| . | . . | | |
| . | . . | | 1. A : ioa{i:l}()
| . | . . | | 2. r : rel()
| . | . . | | 3. rho : Decl
| . | . . | | 4. de : sig()
| . | . . | | 5. e : {[[de]] rho}
| . | . . | | 6. a : (Σ[[A.da]] rho)
| . | . . | | 7. tr : trace_env([[A.da]] rho)
| . | . . | | 8. tc_ioa(A;de)
| . | . . | | 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | | 10. single_valued_decls(A.ds)
| . | . . | | 11. s : {[[A.ds]] rho}
| . | . . | | 12. x' : {[[A.ds]] rho}
| . | . . | | 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | 14. covers_rel(A;r)
| . | . . | | 15. ∀p:pre().
| . | . . | | |   p ∈ A.pre
| . | . . | | |   ⇒
| . | . . | | |   p.kind = kind(a)
| . | . . | | |   ⇒
| . | . . | | |   [[p.rel]] rho A.ds ... de e s value(a) niltrace()
| . | . . | | 16. ∀ef:eff().
| . | . . | | |   ef ∈ A.eff
| . | . . | | |   ⇒
| . | . . | | |   ef.kind = kind(a)
| . | . . | | |   ⇒
| . | . . | | |   x'.ef.smt.lbl
| . | . . | | |   =
| . | . . | | |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | 17. ∀fr:frame().
| . | . . | | |   fr ∈ A.frame
| . | . . | | |   ⇒
| . | . . | | |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | | 18. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | | 19. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | | 20. ∀r@0:rel().
| . | . . | | |   r@0 ∈ col_subst2(λx.
| . | . . | | |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . . | | |   ⇒
| . | . . | | |   [[r@0]] rho A.ds dec_lookup(A.da;...) de e s ... tr
| . | . . | | 21. r' : rel()
| . | . . | | 22. as : (Label×Term) List
| . | . . | | 23. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | | 24. ∀i:ℕ.
| . | . . | | |   i<||as||
| . | . . | | |   ⇒
| . | . . | | |   2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | | 25. r' = rel_subst2(as;r)
| . | . . | | 26. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | 27. i : ℕ||as||
| . | . . | | 28. mentions_trace(2of(as[i]))
```

```
| . | . . | | ⊢  ∃e:eff(). e ∈ A.eff & mentions_trace(e.smt.term)
| . | . . | | | by InstHyp [i] (-5) THENA Auto THEN Reduce (-1)
| . | . . | | |
| . | . . | | 1. A : ioa{i:l}()
| . | . . | | 2. r : rel()
| . | . . | | 3. rho : Decl
| . | . . | | 4. de : sig()
| . | . . | | 5. e : {[[de]] rho}
| . | . . | | 6. a : (Σ[[A.da]] rho)
| . | . . | | 7. tr : trace_env([[A.da]] rho)
| . | . . | | 8. tc_ioa(A;de)
| . | . . | | 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | | 10. single_valued_decls(A.ds)
| . | . . | | 11. s : {[[A.ds]] rho}
| . | . . | | 12. x' : {[[A.ds]] rho}
| . | . . | | 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | 14. covers_rel(A;r)
| . | . . | | 15. ∀p:pre().
| . | . . | | |   p ∈ A.pre
| . | . . | | |   ⇒
| . | . . | | |   p.kind = kind(a)
| . | . . | | |   ⇒
| . | . . | | |   [[p.rel]] rho A.ds ... de e s value(a) niltrace()
| . | . . | | 16. ∀ef:eff().
| . | . . | | |   ef ∈ A.eff
| . | . . | | |   ⇒
| . | . . | | |   ef.kind = kind(a)
| . | . . | | |   ⇒
| . | . . | | |   x'.ef.smt.lbl
| . | . . | | |   =
| . | . . | | |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | 17. ∀fr:frame().
| . | . . | | |   fr ∈ A.frame
| . | . . | | |   ⇒
| . | . . | | |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | | 18. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | | 19. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | | 20. ∀r@0:rel().
| . | . . | | |   r@0 ∈ col_subst2(λx.
| . | . . | | |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . . | | |   ⇒
| . | . . | | |   [[r@0]] rho A.ds dec_lookup(A.da;...) de e s ... tr
| . | . . | | 21. r' : rel()
| . | . . | | 22. as : (Label×Term) List
| . | . . | | 23. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | | 24. ∀i:ℕ.
| . | . . | | |   i<||as||
| . | . . | | |   ⇒
| . | . . | | |   2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | | 25. r' = rel_subst2(as;r)
| . | . . | | 26. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | 27. i : ℕ||as||
| . | . . | | 28. mentions_trace(2of(as[i]))
| . | . . | | 29. 2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | | ⊢  ∃e:eff(). e ∈ A.eff & mentions_trace(e.smt.term)
```

```
| . | . . | | | by Unfold ‘smts_eff‘ (-1) THEN Unfold ‘smt_terms‘ (-1)
| . | . . | | |
| . | . . | | THEN RW ColMemberC (-1) THEN ExRepD
| . | . . | |
| . | . . | | |    THEN Unfold ‘action_effect‘ (-3)
| . | . . | |
| . | . . | | |    THEN RW ColMemberC (-3)
| . | . . | | THEN D (-3) THEN ExRepD
| . | . . | | |
| . | . . | | |\
| . | . . | | | 1. A : ioa{i:l}()
| . | . . | | | 2. r : rel()
| . | . . | | | 3. rho : Decl
| . | . . | | | 4. de : sig()
| . | . . | | | 5. e : {[[de]] rho}
| . | . . | | | 6. a : (Σ[[A.da]] rho)
| . | . . | | | 7. tr : trace_env([[A.da]] rho)
| . | . . | | | 8. tc_ioa(A;de)
| . | . . | | | 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | | | 10. single_valued_decls(A.ds)
| . | . . | | | 11. s : {[[A.ds]] rho}
| . | . . | | | 12. x' : {[[A.ds]] rho}
| . | . . | | | 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | | 14. covers_rel(A;r)
| . | . . | | | 15. ∀p:pre().
| . | . . | | |     p ∈ A.pre
| . | . . | | |     ⇒
| . | . . | | |     p.kind = kind(a)
| . | . . | | |     ⇒
| . | . . | | |     [[p.rel]] rho A.ds ... de e s value(a) niltrace()
| . | . . | | | 16. ∀ef:eff().
| . | . . | | |     ef ∈ A.eff
| . | . . | | |     ⇒
| . | . . | | |     ef.kind = kind(a)
| . | . . | | |     ⇒
| . | . . | | |     x'.ef.smt.lbl
| . | . . | | |     =
| . | . . | | |     [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | | 17. ∀fr:frame().
| . | . . | | |     fr ∈ A.frame
| . | . . | | |     ⇒
| . | . . | | |     ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | | | 18. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | | | 19. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | | | 20. ∀r@0:rel().
| . | . . | | |     r@0 ∈ col_subst2(λx.
| . | . . | | |     smts_eff(action_effect(kind(a);A.eff;...);x);r)
| . | . . | | |     ⇒
| . | . . | | |     [[r@0]] rho A.ds dec_lookup(...;...) de e s ... tr
| . | . . | | | 21. r' : rel()
| . | . . | | | 22. as : (Label×Term) List
| . | . . | | | 23. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | | | 24. ∀i:ℕ.
| . | . . | | |     i<||as||
| . | . . | | |     ⇒
```

```
| . | . . | | |     2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | | | 25. r' = rel_subst2(as;r)
| . | . . | | | 26. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | | 27. i : ℕ||as||
| . | . . | | | 28. mentions_trace(2of(as[i]))
| . | . . | | | 29. s1 : smt()
| . | . . | | | 30. e1 : eff()
| . | . . | | | 31. e1 ∈ A.eff
| . | . . | | | 32. e1.kind =1 kind(a)
| . | . . | | | 33. s1 = e1.smt
| . | . . | | | 34. s1.lbl =1 1of(as[i])
| . | . . | | | 35. 2of(as[i]) = s1.term
| . | . . | | | ⊢  ∃e:eff(). e ∈ A.eff & mentions_trace(e.smt.term)
| . | . . | | |   by InstConcl [e1] THEN Auto
| . | . . | | | THEN SubstFor e1.smt 0
| . | . . | | |
| . | . . | | | THENM SubstFor s1.term 0
| . | . . | | | THEN Auto
| . | . . | | |
| . | . . | | .\
| . | . . | | . 1. A : ioa{i:l}()
| . | . . | | . 2. r : rel()
| . | . . | | . 3. rho : Decl
| . | . . | | . 4. de : sig()
| . | . . | | . 5. e : {[[de]] rho}
| . | . . | | . 6. a : (Σ[[A.da]] rho)
| . | . . | | . 7. tr : trace_env([[A.da]] rho)
| . | . . | | . 8. tc_ioa(A;de)
| . | . . | | . 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | | . 10. single_valued_decls(A.ds)
| . | . . | | . 11. s : {[[A.ds]] rho}
| . | . . | | . 12. x' : {[[A.ds]] rho}
| . | . . | | . 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | . 14. covers_rel(A;r)
| . | . . | | . 15. ∀p:pre().
| . | . . | | . |   p ∈ A.pre
| . | . . | | . |   ⇒
| . | . . | | . |   p.kind = kind(a)
| . | . . | | . |   ⇒
| . | . . | | . |   [[p.rel]] rho A.ds ... de e s value(a) niltrace()
| . | . . | | . 16. ∀ef:eff().
| . | . . | | . |   ef ∈ A.eff
| . | . . | | . |   ⇒
| . | . . | | . |   ef.kind = kind(a)
| . | . . | | . |   ⇒
| . | . . | | . |   x'.ef.smt.lbl
| . | . . | | . |   =
| . | . . | | . |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | . 17. ∀fr:frame().
| . | . . | | . |   fr ∈ A.frame
| . | . . | | . |   ⇒
| . | . . | | . |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | | . 18. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | | . 19. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | | . 20. ∀r@0:rel().
```

```
| . | . . | | . |   r@0 ∈ col_subst2(λx.
| . | . . | | . |   smts_eff(action_effect(kind(a);A.eff;...);x);r)
| . | . . | | . |   ⇒
| . | . . | | . |   [[r@0]] rho A.ds dec_lookup(...;...) de e s ... tr
| . | . . | | . 21. r' : rel()
| . | . . | | . 22. as : (Label×Term) List
| . | . . | | . 23. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | | . 24. ∀i:ℕ.
| . | . . | | . |   i<||as||
| . | . . | | . |   ⇒
| . | . . | | . |   2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | | . 25. r' = rel_subst2(as;r)
| . | . . | | . 26. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | . 27. i : ℕ||as||
| . | . . | | . 28. mentions_trace(2of(as[i]))
| . | . . | | . 29. s1 : smt()
| . | . . | | . 30. f : frame()
| . | . . | | . 31. f ∈ A.frame
| . | . . | | . 32. ¬₂kind(a) ∈ f.acts
| . | . . | | . 33. s1 = mk_smt(f.var, f.var, f.typ)
| . | . . | | . 34. s1.lbl =₁ 1of(as[i])
| . | . . | | . 35. 2of(as[i]) = s1.term
| . | . . | | . ⊢  ∃e:eff(). e ∈ A.eff & mentions_trace(e.smt.term)
| . | . . | | . | by Assert ¬mentions_trace(2of(as[i])) THEN Auto
| . | . . | | .
| . | . . | | . |    THEN SubstFor 2of(as[i]) 0
| . | . . | | . THENM SubstFor s1
| . | . . | | . |     0 THEN Auto
| . | . . | | . |
| . | . . | | . 1. A : ioa{i:l}()
| . | . . | | . 2. r : rel()
| . | . . | | . 3. rho : Decl
| . | . . | | . 4. de : sig()
| . | . . | | . 5. e : {[[de]] rho}
| . | . . | | . 6. a : (Σ[[A.da]] rho)
| . | . . | | . 7. tr : trace_env([[A.da]] rho)
| . | . . | | . 8. tc_ioa(A;de)
| . | . . | | . 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | | . 10. single_valued_decls(A.ds)
| . | . . | | . 11. s : {[[A.ds]] rho}
| . | . . | | . 12. x' : {[[A.ds]] rho}
| . | . . | | . 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | . 14. covers_rel(A;r)
| . | . . | | . 15. ∀p:pre().
| . | . . | | .     p ∈ A.pre
| . | . . | | .     ⇒
| . | . . | | .     p.kind = kind(a)
| . | . . | | .     ⇒
| . | . . | | .     [[p.rel]] rho A.ds ... de e s value(a) niltrace()
| . | . . | | . 16. ∀ef:eff().
| . | . . | | .     ef ∈ A.eff
| . | . . | | .     ⇒
| . | . . | | .     ef.kind = kind(a)
| . | . . | | .     ⇒
| . | . . | | .     x'.ef.smt.lbl
```

```
| . | . . | | .     =
| . | . . | | .     [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | . 17. ∀fr:frame().
| . | . . | | .     fr ∈ A.frame
| . | . . | | .     ⇒
| . | . . | | .     ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | | . 18. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | | . 19. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | | . 20. ∀r@0:rel().
| . | . . | | .     r@0 ∈ col_subst2(λx.
| . | . . | | .     smts_eff(action_effect(kind(a);A.eff;...);x);r)
| . | . . | | .     ⇒
| . | . . | | .     [[r@0]] rho A.ds dec_lookup(...;...) de e s ... tr
| . | . . | | . 21. r' : rel()
| . | . . | | . 22. as : (Label×Term) List
| . | . . | | . 23. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | | . 24. ∀i:ℕ.
| . | . . | | .     i<||as||
| . | . . | | .     ⇒
| . | . . | | .     2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | | . 25. r' = rel_subst2(as;r)
| . | . . | | . 26. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | . 27. i : ℕ||as||
| . | . . | | . 28. mentions_trace(2of(as[i]))
| . | . . | | . 29. s1 : smt()
| . | . . | | . 30. f : frame()
| . | . . | | . 31. f ∈ A.frame
| . | . . | | . 32. ¬₂kind(a) ∈ f.acts
| . | . . | | . 33. s1 = mk_smt(f.var, f.var, f.typ)
| . | . . | | . 34. s1.lbl =₁ 1of(as[i])
| . | . . | | . 35. 2of(as[i]) = s1.term
| . | . . | | . ⊢  ¬mentions_trace(mk_smt(f.var, f.var, f.typ).term)
| . | . . | | .   by Reduce 0 THEN SimpConcl
| . | . . | |
| . | . . | |\
| . | . . | | 1. A : ioa{i:l}()
| . | . . | | 2. r : rel()
| . | . . | | 3. rho : Decl
| . | . . | | 4. de : sig()
| . | . . | | 5. e : {[[de]] rho}

| . | . . | | 6. a : (Σ[[A.da]] rho)
| . | . . | | 7. tr : trace_env([[A.da]] rho)
| . | . . | | 8. tc_ioa(A;de)
| . | . . | | 9. ¬ioa_mentions_trace(A)
| . | . . | | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | | 11. single_valued_decls(A.ds)
| . | . . | | 12. s : {[[A.ds]] rho}
| . | . . | | 13. x' : {[[A.ds]] rho}
| . | . . | | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | 15. covers_rel(A;r)
| . | . . | | 16. ∀p:pre().
| . | . . | | |   p ∈ A.pre
| . | . . | | |   ⇒
| . | . . | | |   p.kind = kind(a)
| . | . . | | |   ⇒
```

```
| . | . . | | |   [[p.rel]] rho A.ds ... de e s value(a) niltrace()
| . | . . | | 17. ∀ef:eff().
| . | . . | | |   ef ∈ A.eff
| . | . . | | |   ⇒
| . | . . | | |   ef.kind = kind(a)
| . | . . | | |   ⇒
| . | . . | | |   x'.ef.smt.lbl
| . | . . | | |   =
| . | . . | | |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | 18. ∀fr:frame().
| . | . . | | |   fr ∈ A.frame
| . | . . | | |   ⇒
| . | . . | | |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | | 19. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | | 20. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | | 21. ∀r@0:rel().
| . | . . | | |   r@0 ∈ col_subst2(λx.
| . | . . | | |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . . | | |   ⇒
| . | . . | | |   [[r@0]] rho A.ds dec_lookup(A.da;...) de e s ... tr
| . | . . | | 22. r' : rel()
| . | . . | | 23. as : (Label×Term) List
| . | . . | | 24. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | | 25. ∀i:ℕ.
| . | . . | | |   i<||as||
| . | . . | | |   ⇒
| . | . . | | |   2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | | 26. r' = rel_subst2(as;r)
| . | . . | | 27. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | ⊢  ∀x:Label.
| . | . . | | |  (x ∈ rel_primed_vars(r))
| . | . . | | |  ⇒
| . | . . | | |  (∀t:SimpleType.
| . | . . | | |   mk_dec(x, t) ∈ A.ds
| . | . . | | |   ⇒
| . | . . | | |   t ∈ term_types(A.ds;dec_lookup(A.da;kind(a));de;...)
| . | . . | | |   & x'.x
| . | . . | | |     =
| . | . . | | |     [[apply_alist(as;x;x)]] 1of(e) s value(a) tr)
| . | . . | | | by MoveToHyp (-10) (-1) THEN
| . | . . | |
| . | . . | | |    ...
| . | . . | | THEN InstLemma `apply_alist_member` [Term;as;x
| . | . . | | |    ;x] THENA Auto
| . | . . | |
| . | . . | | |    THEN Unfold `l_member` (-1) THEN Reduce (-1)
| . | . . | |
| . | . . | | |    THEN ExRepDHyps
| . | . . | | THEN Subst
| . | . . | | |    apply_alist(as;x;x) ~ 2of(as[i]) 0
| . | . . | | THENM Subst
| . | . . | | |    x ~ 1of(as[i]) 0
| . | . . | |
| . | . . | | |    ...
| . | . . | | THEN AllHyps (\i. InstHyp [i
```

```
| . | . . | | |    ] i THENA Complete Auto)
| . | . . | | THEN Reduce (-1)
| . | . . | |
| . | . . | | |    THEN MoveToConcl (-1)
| . | . . | | THEN GenConcl 2of(as[i])
| . | . . | | |                                          =
| . | . . | | |                                          xxx
| . | . . | |
| . | . . | | |    THENM GenConcl 1of(as[i]) = xx
| . | . . | | THENA Auto
| . | . . | |
| . | . . | | |    THEN D 0 THENA Auto•
| . | . . | | |
| . | . . | | 1. A : ioa{i:l}()
| . | . . | | 2. r : rel()
| . | . . | | 3. rho : Decl
| . | . . | | 4. de : sig()
| . | . . | | 5. e : {[[de]] rho}
| . | . . | | 6. a : (Σ[[A.da]] rho)
| . | . . | | 7. tr : trace_env([[A.da]] rho)
| . | . . | | 8. tc_ioa(A;de)
| . | . . | | 9. ¬ioa_mentions_trace(A)
| . | . . | | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | | 11. single_valued_decls(A.ds)
| . | . . | | 12. s : {[[A.ds]] rho}
| . | . . | | 13. x' : {[[A.ds]] rho}
| . | . . | | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | 15. covers_rel(A;r)
| . | . . | | 16. ∀p:pre().
| . | . . | | |   p ∈ A.pre
| . | . . | | |   ⇒
| . | . . | | |   p.kind = kind(a)
| . | . . | | |   ⇒
| . | . . | | |   [[p.rel]] rho A.ds ... de e s value(a) niltrace()
| . | . . | | 17. ∀ef:eff().
| . | . . | | |   ef ∈ A.eff
| . | . . | | |   ⇒
| . | . . | | |   ef.kind = kind(a)
| . | . . | | |   ⇒
| . | . . | | |   x'.ef.smt.lbl
| . | . . | | |   =
| . | . . | | |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | 18. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | 19. ∀fr:frame().
| . | . . | | |   fr ∈ A.frame
| . | . . | | |   ⇒
| . | . . | | |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | | 20. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | | 21. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | | 22. ∀r@0:rel().
| . | . . | | |   r@0 ∈ col_subst2(λx.
| . | . . | | |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . . | | |   ⇒
| . | . . | | |   [[r@0]] rho A.ds dec_lookup(A.da;...) de e s ... tr
| . | . . | | 23. r' : rel()
```

```
| . | . . | | 24. as : (Label×Term) List
| . | . . | | 25. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | | 26. ∀i:ℕ.
| . | . . | | |   i<||as||
| . | . . | | |   ⇒
| . | . . | | |   2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | | 27. r' = rel_subst2(as;r)
| . | . . | | 28. x : Label
| . | . . | | 29. (x ∈ 1of(unzip(as)))
| . | . . | | 30. i : ℕ
| . | . . | | 31. i<||as||
| . | . . | | 32. <x,apply_alist(as;x;x)> = as[i]
| . | . . | | 33. xxx : Term
| . | . . | | 34. 2of(as[i]) = xxx
| . | . . | | 35. xx : Label
| . | . . | | 36. 1of(as[i]) = xx
| . | . . | | 37. xxx ∈ smts_eff(action_effect(kind(a);A.eff;...);xx)
| . | . . | | ⊢  ∀t:SimpleType.
| . | . . | | |  mk_dec(xx, t) ∈ A.ds
| . | . . | | |  ⇒
| . | . . | | |  t ∈ term_types(A.ds;dec_lookup(A.da;kind(a));de;xxx)
| . | . . | | |  & x'.xx = [[xxx]] 1of(e) s value(a) tr
| . | . . | | | by Unfolds ``smts_eff action_effect`` (-1)
| . | . . | |
| . | . . | | |    THEN Unfold `smt_terms` (-1)
| . | . . | |
| . | . . | | |    THEN RW ColMemberC (-1)
| . | . . | | THEN ExRepDHyps
| . | . . | |
| . | . . | | |    THEN HypSubstSq (-1) 0
| . | . . | | THEN Thin (-1)
| . | . . | |
| . | . . | | |    THEN EqLblFwd
| . | . . | | THEN RevHypSubstSq (-1) 0
| . | . . | |
| . | . . | | |    THEN Thin (-1)
| . | . . | | THEN Thin (-1)
| . | . . | | THEN D (-1)
| . | . . | |
| . | . . | | |    THEN ExRepDHyps
| . | . . | | THEN HypSubstSq (-1) 0
| . | . . | |
| . | . . | | |    THEN Reduce 0•
| . | . . | | |
| . | . . | | |\
| . | . . | | | 1. A : ioa{i:l}()
| . | . . | | | 2. r : rel()
| . | . . | | | 3. rho : Decl
| . | . . | | | 4. de : sig()
| . | . . | | | 5. e : {[[de]] rho}
| . | . . | | | 6. a : (Σ[[A.da]] rho)
| . | . . | | | 7. tr : trace_env([[A.da]] rho)
| . | . . | | | 8. tc_ioa(A;de)
| . | . . | | | 9. ¬ioa_mentions_trace(A)
| . | . . | | | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
```

```
| . | . . | | | 11. single_valued_decls(A.ds)
| . | . . | | | 12. s : {[[A.ds]] rho}
| . | . . | | | 13. x' : {[[A.ds]] rho}
| . | . . | | | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | | 15. covers_rel(A;r)
| . | . . | | | 16. ∀p:pre().
| . | . . | | |   p ∈ A.pre
| . | . . | | |   ⇒
| . | . . | | |   p.kind = kind(a)
| . | . . | | |   ⇒
| . | . . | | |   [[p.rel]] rho A.ds ... de e s value(a) niltrace()
| . | . . | | | 17. ∀ef:eff().
| . | . . | | |   ef ∈ A.eff
| . | . . | | |   ⇒
| . | . . | | |   ef.kind = kind(a)
| . | . . | | |   ⇒
| . | . . | | |   x'.ef.smt.lbl
| . | . . | | |   =
| . | . . | | |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | | 18. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | | 19. ∀fr:frame().
| . | . . | | |   fr ∈ A.frame
| . | . . | | |   ⇒
| . | . . | | |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | | | 20. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | | | 21. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | | | 22. ∀r@0:rel().
| . | . . | | |   r@0 ∈ col_subst2(λx.
| . | . . | | |   smts_eff(action_effect(kind(a);A.eff;...);x);r)
| . | . . | | |   ⇒
| . | . . | | |   [[r@0]] rho A.ds dec_lookup(...;...) de e s ... tr
| . | . . | | | 23. r' : rel()
| . | . . | | | 24. as : (Label×Term) List
| . | . . | | | 25. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | | | 26. ∀i:ℕ.
| . | . . | | |   i<||as||
| . | . . | | |   ⇒
| . | . . | | |   2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | | | 27. r' = rel_subst2(as;r)
| . | . . | | | 28. x : Label
| . | . . | | | 29. (x ∈ 1of(unzip(as)))
| . | . . | | | 30. i : ℕ
| . | . . | | | 31. i<||as||
| . | . . | | | 32. <x,apply_alist(as;x;x)> = as[i]
| . | . . | | | 33. xxx : Term
| . | . . | | | 34. 2of(as[i]) = xxx
| . | . . | | | 35. xx : Label
| . | . . | | | 36. 1of(as[i]) = xx
| . | . . | | | 37. s1 : smt()
| . | . . | | | 38. e1 : eff()
| . | . . | | | 39. e1 ∈ A.eff
| . | . . | | | 40. e1.kind =1 kind(a)
| . | . . | | | 41. s1 = e1.smt
| . | . . | | | ⊢  ∀t:SimpleType.
| . | . . | | |  |  mk_dec(e1.smt.lbl, t) ∈ A.ds
```

```
| . | . . | | | |  ⇒
| . | . . | | | |  t ∈ term_types(A.ds;dec_lookup(A.da;...);de;...)
| . | . . | | | |  & x'.e1.smt.lbl
| . | . . | | | |    =
| . | . . | | | |    [[e1.smt.term]] 1of(e) s value(a) tr
| . | . . | | | | by EqLblFwd THEN AllHyps (\i. InstHyp [e1
| . | . . | | | |    ] i THENA Complete Auto) THEN
| . | . . | | | Auto THEN
| . | . . | | |
| . | . . | | | |    ...
| . | . . | | |
| . | . . | | | |    ...e1] (i+1) THENA Auto) THEN
| . | . . | | | Try (Subst
| . | . . | | | |    t ∼ e1.smt.typ 0
| . | . . | | |
| . | . . | | | |    ...
| . | . . | | | |                              THEN InstHyp [
| . | . . | | | |    e1.smt.lbl] i THENA Auto
| . | . . | | |
| . | . . | | | |    ...
| . | . . | | | |           THEN  Auto;
| . | . . | | | |           Auto]).
| . | . . | | | |
| . | . . | | | |
| . | . . | | | |\
| . | . . | | | | 1. A : ioa{i:l}()
| . | . . | | | | 2. r : rel()
| . | . . | | | | 3. rho : Decl
| . | . . | | | | 4. de : sig()
| . | . . | | | | 5. e : {[[de]] rho}
| . | . . | | | | 6. a : (Σ[[A.da]] rho)
| . | . . | | | | 7. tr : trace_env([[A.da]] rho)
| . | . . | | | | 8. tc_pred(A.init;A.ds;<>;de)
| . | . . | | | | 9. ∀p:pre().
| . | . . | | | |    p ∈ A.pre
| . | . . | | | |    ⇒
| . | . . | | | |    tc(p.rel;A.ds;dec_lookup(A.da;p.kind);de)
| . | . . | | | | 10. ∀ef:eff().
| . | . . | | | |     ef ∈ A.eff
| . | . . | | | |     ⇒
| . | . . | | | |     mk_dec(ef.kind, ef.typ) ∈ A.da
| . | . . | | | |     & mk_dec(ef.smt.lbl, ef.smt.typ) ∈ A.ds
| . | . . | | | |     & ef.smt.typ ∈ term_types(A.ds;<ef.typ>;de;...)
| . | . . | | | | 11. ∀f:frame().
| . | . . | | | |     f ∈ A.frame ⇒ mk_dec(f.var, f.typ) ∈ A.ds
| . | . . | | | | 12. ¬ioa_mentions_trace(A)
| . | . . | | | | 13. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | | | | 14. single_valued_decls(A.ds)
| . | . . | | | | 15. s : {[[A.ds]] rho}
| . | . . | | | | 16. x' : {[[A.ds]] rho}
| . | . . | | | | 17. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | | | 18. covers_rel(A;r)
| . | . . | | | | 19. ∀p:pre().
| . | . . | | | |     p ∈ A.pre
| . | . . | | | |     ⇒
```

```
| . | . | . | | | |     p.kind = kind(a)
| . | . | . | | | |     ⇒
| . | . | . | | | |     [[p.rel]] rho A.ds ... de e s value(a) ...
| . | . | . | | | | 20. ∀ef:eff().
| . | . | . | | | |     ef ∈ A.eff
| . | . | . | | | |     ⇒
| . | . | . | | | |     ef.kind = kind(a)
| . | . | . | | | |     ⇒
| . | . | . | | | |     x'.ef.smt.lbl
| . | . | . | | | |     =
| . | . | . | | | |     [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . | . | | | | 21. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . | . | | | | 22. ∀fr:frame().
| . | . | . | | | |     fr ∈ A.frame
| . | . | . | | | |     ⇒
| . | . | . | | | |     ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . | . | | | | 23. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . | . | | | | 24. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . | . | | | | 25. ∀r@0:rel().
| . | . | . | | | |     r@0 ∈ col_subst2(λx.
| . | . | . | | | |     smts_eff(action_effect(kind(a);A.eff;...);x);r)
| . | . | . | | | |     ⇒
| . | . | . | | | |     [[r@0]] rho A.ds ... de e s value(a) tr
| . | . | . | | | | 26. r' : rel()
| . | . | . | | | | 27. as : (Label×Term) List
| . | . | . | | | | 28. 1of(unzip(as)) = rel_primed_vars(r)
| . | . | . | | | | 29. ∀i:ℕ.
| . | . | . | | | |     i<||as||
| . | . | . | | | |     ⇒
| . | . | . | | | |     2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . | . | | | | 30. r' = rel_subst2(as;r)
| . | . | . | | | | 31. x : Label
| . | . | . | | | | 32. (x ∈ 1of(unzip(as)))
| . | . | . | | | | 33. i : ℕ
| . | . | . | | | | 34. i<||as||
| . | . | . | | | | 35. <x,apply_alist(as;x;x)> = as[i]
| . | . | . | | | | 36. xxx : Term
| . | . | . | | | | 37. 2of(as[i]) = xxx
| . | . | . | | | | 38. xx : Label
| . | . | . | | | | 39. 1of(as[i]) = xx
| . | . | . | | | | 40. s1 : smt()
| . | . | . | | | | 41. e1 : eff()
| . | . | . | | | | 42. e1 ∈ A.eff
| . | . | . | | | | 43. e1.kind =₁ kind(a)
| . | . | . | | | | 44. s1 = e1.smt
| . | . | . | | | | 45. e1.kind = kind(a)
| . | . | . | | | | 46. x'.e1.smt.lbl
| . | . | . | | | |     =
| . | . | . | | | |     [[e1.smt.term]] 1of(e) s value(a) niltrace()
| . | . | . | | | | 47. t : SimpleType
| . | . | . | | | | 48. mk_dec(e1.smt.lbl, t) ∈ A.ds
| . | . | . | | | | 49. mk_dec(e1.kind, e1.typ) ∈ A.da
| . | . | . | | | | 50. mk_dec(e1.smt.lbl, e1.smt.typ) ∈ A.ds
| . | . | . | | | | 51. e1.smt.typ ∈ term_types(A.ds;<e1.typ>;de;...)
| . | . | . | | | | ⊢  e1.smt.typ ∈ term_types(A.ds;...;de;e1.smt.term)
```

```
| . | . . | | | |   by Using [‘ds1’,A.ds; ‘da1’,<e1.typ>
| . | . . | | | |      ...
| . | . . | | | |
| . | . . | | | |      ...
| . | . . | | | |
| . | . . | | | |      ...
| . | . . | | | | THEN Try (SubstFor kind(a)
| . | . . | | | |       0 THENM SubstFor x1  0 THEN Auto)·
| . | . . | | | |
| . | . . | | | .\
| . | . . | | | . 1. A : ioa{i:l}()
| . | . . | | | . 2. r : rel()
| . | . . | | | . 3. rho : Decl
| . | . . | | | . 4. de : sig()
| . | . . | | | . 5. e : {[[de]] rho}
| . | . . | | | . 6. a : (Σ[[A.da]] rho)
| . | . . | | | . 7. tr : trace_env([[A.da]] rho)
| . | . . | | | . 8. tc_pred(A.init;A.ds;<>;de)
| . | . . | | | . 9. ∀p:pre().
| . | . . | | | . |  p ∈ A.pre
| . | . . | | | . |  ⇒
| . | . . | | | . |  tc(p.rel;A.ds;dec_lookup(A.da;p.kind);de)
| . | . . | | | . 10. ∀ef:eff().
| . | . . | | | . |   ef ∈ A.eff
| . | . . | | | . |   ⇒
| . | . . | | | . |   mk_dec(ef.kind, ef.typ) ∈ A.da
| . | . . | | | . |   & mk_dec(ef.smt.lbl, ef.smt.typ) ∈ A.ds
| . | . . | | | . |   & ef.smt.typ ∈ term_types(A.ds;<ef.typ>;de;...)
| . | . . | | | . 11. ∀f:frame().
| . | . . | | | . |   f ∈ A.frame ⇒ mk_dec(f.var, f.typ) ∈ A.ds
| . | . . | | | . 12. ¬ioa_mentions_trace(A)
| . | . . | | | . 13. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | | | . 14. single_valued_decls(A.ds)
| . | . . | | | . 15. s : {[[A.ds]] rho}
| . | . . | | | . 16. x’ : {[[A.ds]] rho}
| . | . . | | | . 17. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | | . 18. covers_rel(A;r)
| . | . . | | | . 19. ∀p:pre().
| . | . . | | | . |   p ∈ A.pre
| . | . . | | | . |   ⇒
| . | . . | | | . |   p.kind = kind(a)
| . | . . | | | . |   ⇒
| . | . . | | | . |   [[p.rel]] rho A.ds ... de e s value(a) ...
| . | . . | | | . 20. ∀ef:eff().
| . | . . | | | . |   ef ∈ A.eff
| . | . . | | | . |   ⇒
| . | . . | | | . |   ef.kind = kind(a)
| . | . . | | | . |   ⇒
| . | . . | | | . |   x’.ef.smt.lbl
| . | . . | | | . |   =
| . | . . | | | . |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | | . 21. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | | . 22. ∀fr:frame().
| . | . . | | | . |   fr ∈ A.frame
| . | . . | | | . |   ⇒
```

```
| . | . . | | | . |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | | | . 23. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | | | . 24. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | | | . 25. ∀r@0:rel().
| . | . . | | | . |   r@0 ∈ col_subst2(λx.
| . | . . | | | . |   smts_eff(action_effect(kind(a);A.eff;...);x);r)
| . | . . | | | . |   ⇒
| . | . . | | | . |   [[r@0]] rho A.ds ... de e s value(a) tr
| . | . . | | | . 26. r' : rel()
| . | . . | | | . 27. as : (Label×Term) List
| . | . . | | | . 28. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | | | . 29. ∀i:ℕ.
| . | . . | | | . |   i<||as||
| . | . . | | | . |   ⇒
| . | . . | | | . |   2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | | | . 30. r' = rel_subst2(as;r)
| . | . . | | | . 31. x : Label
| . | . . | | | . 32. (x ∈ 1of(unzip(as)))
| . | . . | | | . 33. i : ℕ
| . | . . | | | . 34. i<||as||
| . | . . | | | . 35. <x,apply_alist(as;x;x)> = as[i]
| . | . . | | | . 36. xxx : Term
| . | . . | | | . 37. 2of(as[i]) = xxx
| . | . . | | | . 38. xx : Label
| . | . . | | | . 39. 1of(as[i]) = xx
| . | . . | | | . 40. s1 : smt()
| . | . . | | | . 41. e1 : eff()
| . | . . | | | . 42. e1 ∈ A.eff
| . | . . | | | . 43. e1.kind =1 kind(a)
| . | . . | | | . 44. s1 = e1.smt
| . | . . | | | . 45. e1.kind = kind(a)
| . | . . | | | . 46. x'.e1.smt.lbl
| . | . . | | | . |   =
| . | . . | | | . |   [[e1.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | | . 47. t : SimpleType
| . | . . | | | . 48. mk_dec(e1.smt.lbl, t) ∈ A.ds
| . | . . | | | . 49. mk_dec(e1.kind, e1.typ) ∈ A.da
| . | . . | | | . 50. mk_dec(e1.smt.lbl, e1.smt.typ) ∈ A.ds
| . | . . | | | . 51. e1.smt.typ ∈ term_types(A.ds;<e1.typ>;de;...)
| . | . . | | | . ⊢  x'.e1.smt.lbl
| . | . . | | | . |  =
| . | . . | | | . |  [[e1.smt.term]] 1of(e) s value(a) tr
| . | . . | | | . | by Subst
| . | . . | | | . |    [[e1.smt.term]] 1of(e) s value(a) tr ~ ... 0
| . | . . | | | .
| . | . . | | | . |       THEN Try Trivial
| . | . . | | | . |
| . | . . | | | . .....equality .....
| . | . . | | | . 1. A : ioa{i:l}()
| . | . . | | | . 2. r : rel()
| . | . . | | | . 3. rho : Decl
| . | . . | | | . 4. de : sig()
| . | . . | | | . 5. e : {[[de]] rho}
| . | . . | | | . 6. a : (Σ[[A.da]] rho)
| . | . . | | | . 7. tr : trace_env([[A.da]] rho)
```

```
| . | . . | | | . 8. tc_pred(A.init;A.ds;<>;de)
| . | . . | | | . 9. ∀p:pre().
| . | . . | | | .    p ∈ A.pre
| . | . . | | | .    ⇒
| . | . . | | | .    tc(p.rel;A.ds;dec_lookup(A.da;p.kind);de)
| . | . . | | | . 10. ∀ef:eff().
| . | . . | | | .     ef ∈ A.eff
| . | . . | | | .     ⇒
| . | . . | | | .     mk_dec(ef.kind, ef.typ) ∈ A.da
| . | . . | | | .     & mk_dec(ef.smt.lbl, ef.smt.typ) ∈ A.ds
| . | . . | | | .     & ef.smt.typ ∈ term_types(A.ds;<ef.typ>;de;...)
| . | . . | | | . 11. ∀f:frame().
| . | . . | | | .     f ∈ A.frame ⇒ mk_dec(f.var, f.typ) ∈ A.ds
| . | . . | | | . 12. ¬ioa_mentions_trace(A)
| . | . . | | | . 13. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | | | . 14. single_valued_decls(A.ds)
| . | . . | | | . 15. s : {[[A.ds]] rho}
| . | . . | | | . 16. x' : {[[A.ds]] rho}
| . | . . | | | . 17. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | | . 18. covers_rel(A;r)
| . | . . | | | . 19. ∀p:pre().
| . | . . | | | .     p ∈ A.pre
| . | . . | | | .     ⇒
| . | . . | | | .     p.kind = kind(a)
| . | . . | | | .     ⇒
| . | . . | | | .     [[p.rel]] rho A.ds ... de e s value(a) ...
| . | . . | | | . 20. ∀ef:eff().
| . | . . | | | .     ef ∈ A.eff
| . | . . | | | .     ⇒
| . | . . | | | .     ef.kind = kind(a)
| . | . . | | | .     ⇒
| . | . . | | | .     x'.ef.smt.lbl
| . | . . | | | .     =
| . | . . | | | .     [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | | . 21. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | | . 22. ∀fr:frame().
| . | . . | | | .     fr ∈ A.frame
| . | . . | | | .     ⇒
| . | . . | | | .     ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | | | . 23. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | | | . 24. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | | | . 25. ∀r@0:rel().
| . | . . | | | .     r@0 ∈ col_subst2(λx.
| . | . . | | | .     smts_eff(action_effect(kind(a);A.eff;...);x);r)
| . | . . | | | .     ⇒
| . | . . | | | .     [[r@0]] rho A.ds ... de e s value(a) tr
| . | . . | | | . 26. r' : rel()
| . | . . | | | . 27. as : (Label×Term) List
| . | . . | | | . 28. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | | | . 29. ∀i:ℕ.
| . | . . | | | .     i<||as||
| . | . . | | | .     ⇒
| . | . . | | | .     2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | | | . 30. r' = rel_subst2(as;r)
| . | . . | | | . 31. x : Label
```

```
| . | . . | | | . 32. (x ∈ 1of(unzip(as)))
| . | . . | | | . 33. i : ℕ
| . | . . | | | . 34. i<||as||
| . | . . | | | . 35. <x,apply_alist(as;x;x)> = as[i]
| . | . . | | | . 36. xxx : Term
| . | . . | | | . 37. 2of(as[i]) = xxx
| . | . . | | | . 38. xx : Label
| . | . . | | | . 39. 1of(as[i]) = xx
| . | . . | | | . 40. s1 : smt()
| . | . . | | | . 41. e1 : eff()
| . | . . | | | . 42. e1 ∈ A.eff
| . | . . | | | . 43. e1.kind =1 kind(a)
| . | . . | | | . 44. s1 = e1.smt
| . | . . | | | . 45. e1.kind = kind(a)
| . | . . | | | . 46. x'.e1.smt.lbl
| . | . . | | | .     =
| . | . . | | | .     [[e1.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | | . 47. t : SimpleType
| . | . . | | | . 48. mk_dec(e1.smt.lbl, t) ∈ A.ds
| . | . . | | | . 49. mk_dec(e1.kind, e1.typ) ∈ A.da
| . | . . | | | . 50. mk_dec(e1.smt.lbl, e1.smt.typ) ∈ A.ds
| . | . . | | | . 51. e1.smt.typ ∈ term_types(A.ds;<e1.typ>;de;...)
| . | . . | | | . ⊢  [[e1.smt.term]] 1of(e) s value(a) tr ∼ ...
| . | . . | | | .   by ...
| . | . . | | | .
| . | . . | | | .      ...
| . | . . | | |
| . | . . | | .\
| . | . . | | . 1. A : ioa{i:l}()
| . | . . | | . 2. r : rel()
| . | . . | | . 3. rho : Decl
| . | . . | | . 4. de : sig()
| . | . . | | . 5. e : {[[de]] rho}
| . | . . | | . 6. a : (Σ[[A.da]] rho)
| . | . . | | . 7. tr : trace_env([[A.da]] rho)
| . | . . | | . 8. tc_ioa(A;de)
| . | . . | | . 9. ¬ioa_mentions_trace(A)
| . | . . | | . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | | . 11. single_valued_decls(A.ds)
| . | . . | | . 12. s : {[[A.ds]] rho}
| . | . . | | . 13. x' : {[[A.ds]] rho}
| . | . . | | . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | . 15. covers_rel(A;r)
| . | . . | | . 16. ∀p:pre().
| . | . . | | . |   p ∈ A.pre
| . | . . | | . |   ⇒
| . | . . | | . |   p.kind = kind(a)
| . | . . | | . |   ⇒
| . | . . | | . |   [[p.rel]] rho A.ds ... de e s value(a) niltrace()
| . | . . | | . 17. ∀ef:eff().
| . | . . | | . |   ef ∈ A.eff
| . | . . | | . |   ⇒
| . | . . | | . |   ef.kind = kind(a)
| . | . . | | . |   ⇒
| . | . . | | . |   x'.ef.smt.lbl
```

```
| . | . . | | . |   =
| . | . . | | . |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | . 18. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | . 19. ∀fr:frame().
| . | . . | | . |   fr ∈ A.frame
| . | . . | | . |   ⇒
| . | . . | | . |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | | . 20. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | | . 21. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | | . 22. ∀r@0:rel().
| . | . . | | . |   r@0 ∈ col_subst2(λx.
| . | . . | | . |   smts_eff(action_effect(kind(a);A.eff;...);x);r)
| . | . . | | . |   ⇒
| . | . . | | . |   [[r@0]] rho A.ds dec_lookup(...;...) de e s ... tr
| . | . . | | . 23. r' : rel()
| . | . . | | . 24. as : (Label×Term) List
| . | . . | | . 25. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | | . 26. ∀i:ℕ.
| . | . . | | . |   i<||as||
| . | . . | | . |   ⇒
| . | . . | | . |   2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | | . 27. r' = rel_subst2(as;r)
| . | . . | | . 28. x : Label
| . | . . | | . 29. (x ∈ 1of(unzip(as)))
| . | . . | | . 30. i : ℕ
| . | . . | | . 31. i<||as||
| . | . . | | . 32. <x,apply_alist(as;x;x)> = as[i]
| . | . . | | . 33. xxx : Term
| . | . . | | . 34. 2of(as[i]) = xxx
| . | . . | | . 35. xx : Label
| . | . . | | . 36. 1of(as[i]) = xx
| . | . . | | . 37. s1 : smt()
| . | . . | | . 38. f : frame()
| . | . . | | . 39. f ∈ A.frame
| . | . . | | . 40. ¬₂kind(a) ∈ f.acts

| . | . . | | . 41. s1 = mk_smt(f.var, f.var, f.typ)
| . | . . | | . ⊢  ∀t:SimpleType.
| . | . . | | . |  mk_dec(f.var, t) ∈ A.ds
| . | . . | | . |  ⇒
| . | . . | | . |  t ∈ dec_lookup(A.ds;f.var) & x'.f.var = s.f.var
| . | . . | | . | by ...
| . | . . | | . THEN AllHyps (\i. InstHyp [f
| . | . . | | . |    ] i THENA Complete Auto)
| . | . . | | . THEN Auto
| . | . . | | .
| . | . . | | . |    ...
| . | . . | | . •
| . | . . | | . |
| . | . . | | . 1. A : ioa{i:l}()
| . | . . | | . 2. r : rel()
| . | . . | | . 3. rho : Decl
| . | . . | | . 4. de : sig()
| . | . . | | . 5. e : {[[de]] rho}
| . | . . | | . 6. a : (Σ[[A.da]] rho)
| . | . . | | . 7. tr : trace_env([[A.da]] rho)
```

```
| . | . . | | . 8. tc_ioa(A;de)
| . | . . | | . 9. ¬ioa_mentions_trace(A)
| . | . . | | . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | | . 11. single_valued_decls(A.ds)
| . | . . | | . 12. s : {[[A.ds]] rho}
| . | . . | | . 13. x' : {[[A.ds]] rho}
| . | . . | | . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | . 15. covers_rel(A;r)
| . | . . | | . 16. ∀p:pre().
| . | . . | | . |   p ∈ A.pre
| . | . . | | . |   ⇒
| . | . . | | . |   p.kind = kind(a)
| . | . . | | . |   ⇒
| . | . . | | . |   [[p.rel]] rho A.ds ... de e s value(a) niltrace()
| . | . . | | . 17. ∀ef:eff().
| . | . . | | . |   ef ∈ A.eff
| . | . . | | . |   ⇒
| . | . . | | . |   ef.kind = kind(a)
| . | . . | | . |   ⇒
| . | . . | | . |   x'.ef.smt.lbl
| . | . . | | . |   =
| . | . . | | . |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | . 18. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | . 19. ∀fr:frame().
| . | . . | | . |   fr ∈ A.frame
| . | . . | | . |   ⇒
| . | . . | | . |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | | . 20. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | | . 21. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | | . 22. ∀r@0:rel().
| . | . . | | . |   r@0 ∈ col_subst2(λx.
| . | . . | | . |   smts_eff(action_effect(kind(a);A.eff;...);x);r)
| . | . . | | . |   ⇒
| . | . . | | . |   [[r@0]] rho A.ds dec_lookup(...;...) de e s ... tr
| . | . . | | . 23. r' : rel()
| . | . . | | . 24. as : (Label×Term) List
| . | . . | | . 25. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | | . 26. ∀i:ℕ.
| . | . . | | . |   i<||as||
| . | . . | | . |   ⇒
| . | . . | | . |   2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | | . 27. r' = rel_subst2(as;r)
| . | . . | | . 28. x : Label
| . | . . | | . 29. (x ∈ 1of(unzip(as)))
| . | . . | | . 30. i : ℕ
| . | . . | | . 31. i<||as||
| . | . . | | . 32. <x,apply_alist(as;x;x)> = as[i]
| . | . . | | . 33. xxx : Term
| . | . . | | . 34. 2of(as[i]) = xxx
| . | . . | | . 35. xx : Label
| . | . . | | . 36. 1of(as[i]) = xx
| . | . . | | . 37. s1 : smt()
| . | . . | | . 38. f : frame()
| . | . . | | . 39. f ∈ A.frame
| . | . . | | . 40. ¬(kind(a) ∈ f.acts)
```

```
| . | . . | | . 41. s1 = mk_smt(f.var, f.var, f.typ)
| . | . . | | . 42. x'.f.var = s.f.var
| . | . . | | . 43. t : SimpleType
| . | . . | | . 44. mk_dec(f.var, t) ∈ A.ds
| . | . . | | . ⊢  x'.f.var = s.f.var
| . | . . | | . | by Try (Fold `r_select` 0) THEN Subst t ∼ f.typ
| . | . . | | . |      0 THEN Auto
| . | . . | | . •
| . | . . | | . |
| . | . . | | . 1. A : ioa{i:l}()
| . | . . | | . 2. r : rel()
| . | . . | | . 3. rho : Decl
| . | . . | | . 4. de : sig()
| . | . . | | . 5. e : {[[de]] rho}
| . | . . | | . 6. a : (Σ[[A.da]] rho)
| . | . . | | . 7. tr : trace_env([[A.da]] rho)
| . | . . | | . 8. tc_ioa(A;de)
| . | . . | | . 9. ¬ioa_mentions_trace(A)
| . | . . | | . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | | . 11. single_valued_decls(A.ds)
| . | . . | | . 12. s : {[[A.ds]] rho}
| . | . . | | . 13. x' : {[[A.ds]] rho}
| . | . . | | . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | . 15. covers_rel(A;r)
| . | . . | | . 16. ∀p:pre().
| . | . . | | . |   p ∈ A.pre
| . | . . | | . |   ⇒
| . | . . | | . |   p.kind = kind(a)
| . | . . | | . |   ⇒
| . | . . | | . |   [[p.rel]] rho A.ds ... de e s value(a) niltrace()
| . | . . | | . 17. ∀ef:eff().
| . | . . | | . |   ef ∈ A.eff
| . | . . | | . |   ⇒
| . | . . | | . |   ef.kind = kind(a)
| . | . . | | . |   ⇒
| . | . . | | . |   x'.ef.smt.lbl
| . | . . | | . |   =
| . | . . | | . |   [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | . 18. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | . 19. ∀fr:frame().
| . | . . | | . |   fr ∈ A.frame
| . | . . | | . |   ⇒
| . | . . | | . |   ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | | . 20. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | | . 21. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | | . 22. ∀r@0:rel().
| . | . . | | . |   r@0 ∈ col_subst2(λx.
| . | . . | | . |   smts_eff(action_effect(kind(a);A.eff;...);x);r)
| . | . . | | . |   ⇒
| . | . . | | . |   [[r@0]] rho A.ds dec_lookup(...;...) de e s ... tr
| . | . . | | . 23. r' : rel()
| . | . . | | . 24. as : (Label×Term) List
| . | . . | | . 25. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | | . 26. ∀i:ℕ.
| . | . . | | . |   i<||as||
```

```
| . | . . | | . |   ⇒
| . | . . | | . |   2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | | . 27. r' = rel_subst2(as;r)
| . | . . | | . 28. x : Label
| . | . . | | . 29. (x ∈ 1of(unzip(as)))
| . | . . | | . 30. i : ℕ
| . | . . | | . 31. i<||as||
| . | . . | | . 32. <x,apply_alist(as;x;x)> = as[i]
| . | . . | | . 33. xxx : Term
| . | . . | | . 34. 2of(as[i]) = xxx
| . | . . | | . 35. xx : Label
| . | . . | | . 36. 1of(as[i]) = xx
| . | . . | | . 37. s1 : smt()
| . | . . | | . 38. f : frame()
| . | . . | | . 39. f ∈ A.frame
| . | . . | | . 40. ¬(kind(a) ∈ f.acts)
| . | . . | | . 41. s1 = mk_smt(f.var, f.var, f.typ)
| . | . . | | . 42. x'.f.var = s.f.var
| . | . . | | . 43. t : SimpleType
| . | . . | | . 44. mk_dec(f.var, t) ∈ A.ds
| . | . . | | . ⊢  t = f.typ
| . | . . | | . | by ...f.var] i THENA Auto
| . | . . | | .
| . | . . | | . |             THEN BackThruHyp (-1) THEN Auto)•
| . | . . | | . |
| . | . . | | . 1. A : ioa{i:l}()
| . | . . | | . 2. r : rel()
| . | . . | | . 3. rho : Decl
| . | . . | | . 4. de : sig()
| . | . . | | . 5. e : {[[de]] rho}
| . | . . | | . 6. a : (Σ[[A.da]] rho)
| . | . . | | . 7. tr : trace_env([[A.da]] rho)
| . | . . | | . 8. tc_ioa(A;de)
| . | . . | | . 9. ¬ioa_mentions_trace(A)
| . | . . | | . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | | . 11. ∀x:Label, t1,t2:SimpleType.
| . | . . | | .     mk_dec(x, t1) ∈ A.ds
| . | . . | | .     ⇒
| . | . . | | .     mk_dec(x, t2) ∈ A.ds ⇒ t1 = t2
| . | . . | | . 12. s : {[[A.ds]] rho}
| . | . . | | . 13. x' : {[[A.ds]] rho}
| . | . . | | . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | . 15. covers_rel(A;r)
| . | . . | | . 16. ∀p:pre().
| . | . . | | .     p ∈ A.pre
| . | . . | | .     ⇒
| . | . . | | .     p.kind = kind(a)
| . | . . | | .     ⇒
| . | . . | | .     [[p.rel]] rho A.ds ... de e s value(a) niltrace()
| . | . . | | . 17. ∀ef:eff().
| . | . . | | .     ef ∈ A.eff
| . | . . | | .     ⇒
| . | . . | | .     ef.kind = kind(a)
| . | . . | | .     ⇒
| . | . . | | .     x'.ef.smt.lbl
```

```
| . | . . | | .     =
| . | . . | | .     [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | | . 18. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | | . 19. ∀fr:frame().
| . | . . | | .     fr ∈ A.frame
| . | . . | | .     ⇒
| . | . . | | .     ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | | . 20. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | | . 21. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | | . 22. ∀r@0:rel().
| . | . . | | .     r@0 ∈ col_subst2(λx.
| . | . . | | .     smts_eff(action_effect(kind(a);A.eff;...);x);r)
| . | . . | | .     ⇒
| . | . . | | .     [[r@0]] rho A.ds dec_lookup(...;...) de e s ... tr
| . | . . | | . 23. r' : rel()
| . | . . | | . 24. as : (Label×Term) List
| . | . . | | . 25. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | | . 26. ∀i:ℕ.
| . | . . | | .     i<||as||
| . | . . | | .     ⇒
| . | . . | | .     2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | | . 27. r' = rel_subst2(as;r)
| . | . . | | . 28. x : Label
| . | . . | | . 29. (x ∈ 1of(unzip(as)))
| . | . . | | . 30. i : ℕ
| . | . . | | . 31. i<||as||
| . | . . | | . 32. <x,apply_alist(as;x;x)> = as[i]
| . | . . | | . 33. xxx : Term
| . | . . | | . 34. 2of(as[i]) = xxx
| . | . . | | . 35. xx : Label
| . | . . | | . 36. 1of(as[i]) = xx
| . | . . | | . 37. s1 : smt()
| . | . . | | . 38. f : frame()
| . | . . | | . 39. f ∈ A.frame
| . | . . | | . 40. ¬(kind(a) ∈ f.acts)
| . | . . | | . 41. s1 = mk_smt(f.var, f.var, f.typ)
| . | . . | | . 42. x'.f.var = s.f.var
| . | . . | | . 43. t : SimpleType
| . | . . | | . 44. mk_dec(f.var, t) ∈ A.ds
| . | . . | | . 45. ∀t1,t2:SimpleType.
| . | . . | | .     mk_dec(f.var, t1) ∈ A.ds
| . | . . | | .     ⇒
| . | . . | | .     mk_dec(f.var, t2) ∈ A.ds ⇒ t1 = t2
| . | . . | | . ⊢  mk_dec(f.var, f.typ) ∈ A.ds
| . | . . | | .   by ...•
| . | . . | |
| . | . . | .\
| . | . . | . 1. A : ioa{i:l}()
| . | . . | . 2. r : rel()
| . | . . | . 3. rho : Decl
| . | . . | . 4. de : sig()
| . | . . | . 5. e : {[[de]] rho}
| . | . . | . 6. a : (Σ[[A.da]] rho)
| . | . . | . 7. tr : trace_env([[A.da]] rho)
| . | . . | . 8. tc_ioa(A;de)
```

```
| . | . . | . 9. ¬ioa_mentions_trace(A)
| . | . . | . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . | . 11. single_valued_decls(A.ds)
| . | . . | . 12. s : {[[A.ds]] rho}
| . | . . | . 13. x' : {[[A.ds]] rho}
| . | . . | . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | . 15. covers_rel(A;r)
| . | . . | . 16. ∀p:pre().
| . | . . | .     p ∈ A.pre
| . | . . | .     ⇒
| . | . . | .     p.kind = kind(a)
| . | . . | .     ⇒
| . | . . | .     [[p.rel]] rho A.ds ... de e s value(a) niltrace()
| . | . . | . 17. ∀ef:eff().
| . | . . | .     ef ∈ A.eff
| . | . . | .     ⇒
| . | . . | .     ef.kind = kind(a)
| . | . . | .     ⇒
| . | . . | .     x'.ef.smt.lbl
| . | . . | .     =
| . | . . | .     [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . | . 18. ∀fr:frame().
| . | . . | .     fr ∈ A.frame
| . | . . | .     ⇒
| . | . . | .     ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . | . 19. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . | . . | . 20. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . | . 21. ∀r@0:rel().
| . | . . | .     r@0 ∈ col_subst2(λx.
| . | . . | .     smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . . | .     ⇒
| . | . . | .     [[r@0]] rho A.ds dec_lookup(A.da;...) de e s ... tr
| . | . . | . 22. r' : rel()
| . | . . | . 23. as : (Label×Term) List
| . | . . | . 24. 1of(unzip(as)) = rel_primed_vars(r)
| . | . . | . 25. ∀i:ℕ.
| . | . . | .     i<||as||
| . | . . | .     ⇒
| . | . . | .     2of(as[i]) ∈ smts_eff(...;1of(as[i]))
| . | . . | . 26. r' = rel_subst2(as;r)
| . | . . | . 27. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | . 28. tc(rel_subst2(as;r);A.ds;dec_lookup(A.da;...);de)
| . | . . | .     & trace_consistent_rel(rho;A.da;tr.proj;...)
| . | . . | .     & ([[rel_subst2(as;r)]] rho A.ds ... de e s ... tr
| . | . . | .        ⟺
| . | . . | .        rel_mng_2(r;
| . | . . | .                  rho;
| . | . . | .                  A.ds;
| . | . . | .                  dec_lookup(A.da;kind(a));
| . | . . | .                  de;
| . | . . | .                  e;
| . | . . | .                  s;
| . | . . | .                  x';
| . | . . | .                  value(a);
| . | . . | .                  tr))
```

```
| . | . . | . ⊢  tc(rel_subst2(as;r);A.ds;dec_lookup(A.da;kind(a));de)
| . | . . | .    & trace_consistent_rel(rho;A.da;tr.proj;...)
| . | . . | .    & ([[rel_subst2(as;r)]] rho A.ds ... de e s ... tr
| . | . . | .       ⇒
| . | . . | .       rel_mng_2(r;
| . | . . | .                 rho;
| . | . . | .                 A.ds;
| . | . . | .                 dec_lookup(A.da;kind(a));
| . | . . | .                 de;
| . | . . | .                 e;
| . | . . | .                 s;
| . | . . | .                 x';
| . | . . | .                 value(a);
| . | . . | .                 tr))
| . | . . | .   by ExRepD THEN D 0 THEN Try (D 0 THEN Trivial)
| . | . . | .
| . | . . | .      THEN Try ( D (-2) THEN Trivial)
| . | . . | .  •
| . | . . |
| . | . . .\
| . | . . . .....subterm .....
| . | . . . 1. A : ioa{i:l}()
| . | . . . 2. r : rel()
| . | . . . 3. rho : Decl
| . | . . . 4. de : sig()
| . | . . . 5. e : {[[de]] rho}
| . | . . . 6. a : (Σ[[A.da]] rho)
| . | . . . 7. tr : trace_env([[A.da]] rho)
| . | . . . 8. tc_ioa(A;de)
| . | . . . 9. ¬ioa_mentions_trace(A)
| . | . . . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . | . . . 11. single_valued_decls(A.ds)
| . | . . . 12. s : {[[A.ds]] rho}
| . | . . . 13. x' : {[[A.ds]] rho}
| . | . . . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . . 15. covers_rel(A;r)
| . | . . . 16. ∀p:pre().
| . | . . .     p ∈ A.pre
| . | . . .     ⇒
| . | . . .     p.kind = kind(a)
| . | . . .     ⇒
| . | . . .     [[p.rel]] rho A.ds dec_lookup(A.da;...) de e s ... ...
| . | . . . 17. ∀ef:eff().
| . | . . .     ef ∈ A.eff
| . | . . .     ⇒
| . | . . .     ef.kind = kind(a)
| . | . . .     ⇒
| . | . . .     x'.ef.smt.lbl
| . | . . .     =
| . | . . .     [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . | . . . 18. ∀fr:frame().
| . | . . .     fr ∈ A.frame
| . | . . .     ⇒
| . | . . .     ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . | . . . 19. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
```

```
| . | . . . 20. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . | . . . 21. ∀r@0:rel().
| . | . . .     r@0 ∈ col_subst2(λx.
| . | . . .     smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . . .     ⇒
| . | . . .     [[r@0]] rho A.ds dec_lookup(A.da;...) de e s ... tr
| . | . . . 22. r' : rel()
| . | . . . 23. r' ∈ col_subst2(λx.
| . | . . .     smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . . . 24. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . | . . . 25. rel()
| . | . . . ⊢  col_subst2(λx.
| . | . . .    smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . | . . .    ∈ Collection(rel())  by Fold ‘pred‘ 0 THEN Auto
| . |
| . .\
| . . 1. A : ioa{i:l}()
| . . 2. r : rel()
| . . 3. rho : Decl
| . . 4. de : sig()
| . . 5. e : {[[de]] rho}
| . . 6. a : (Σ[[A.da]] rho)
| . . 7. tr : trace_env([[A.da]] rho)
| . . 8. tc_ioa(A;de)
| . . 9. ¬ioa_mentions_trace(A)
| . . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . . 11. single_valued_decls(A.ds)
| . . 12. s : [[A]] rho de e.state
| . . 13. x' : [[A]] rho de e.state
| . . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . . 15. covers_rel(A;r)
| . . 16. ∀p:pre().
| . . |   p ∈ A.pre
| . . |   ⇒
| . . |   p.kind = kind(a)
| . . |   ⇒
| . . |   [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . . 17. ∀ef:eff().
| . . |   ef ∈ A.eff
| . . |   ⇒
| . . |   ef.kind = kind(a)
| . . |   ⇒
| . . |   x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . . 18. ∀fr:frame().
| . . |   fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . . 19. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
| . . 20. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . . 21. ∀r@0:rel().
| . . |   r@0 ∈ col_subst2(λx.
| . . |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . . |   ⇒
| . . |   [[r@0]] rho A.ds dec_lookup(A.da;kind(a)) de e s value(a) tr
| . . 22. ∃r':rel().
| . . |   r' ∈ col_subst2(λx.
| . . |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
```

```
| . . |   & tc(r';A.ds;dec_lookup(A.da;kind(a));de)
| . . |     & trace_consistent_rel(rho;A.da;tr.proj;r')
| . . |     & ([[r']] rho A.ds dec_lookup(A.da;kind(a)) de e s ... tr
| . . |        ⇒
| . . |        rel_mng_2(r;
| . . |                  rho;
| . . |                  A.ds;
| . . |                  dec_lookup(A.da;kind(a));
| . . |                  de;
| . . |                  e;
| . . |                  s;
| . . |                  x';
| . . |                  value(a);
| . . |                  tr))
| . . ⊢  rel_mng_2(r;
| . . |            rho;
| . . |            A.ds;
| . . |            dec_lookup(A.da;kind(a));
| . . |            de;
| . . |            e;
| . . |            s;
| . . |            x';
| . . |            value(a);
| . . |            tr)  by ExRepD THEN InstHyp [r'] (-6) THEN Auto•
| . . |
| . . 1. A : ioa{i:l}()
| . . 2. r : rel()
| . . 3. rho : Decl
| . . 4. de : sig()
| . . 5. e : {[[de]] rho}
| . . 6. a : (Σ[[A.da]] rho)
| . . 7. tr : trace_env([[A.da]] rho)
| . . 8. tc_ioa(A;de)
| . . 9. ¬ioa_mentions_trace(A)
| . . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| . . 11. single_valued_decls(A.ds)
| . . 12. s : [[A]] rho de e.state
| . . 13. x' : [[A]] rho de e.state
| . . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . . 15. covers_rel(A;r)
| . . 16. ∀p:pre().
| . .     p ∈ A.pre
| . .     ⇒
| . .     p.kind = kind(a)
| . .     ⇒
| . .     [[p.rel]] rho A.ds dec_lookup(A.da;kind(a)) de e s ... ...
| . . 17. ∀ef:eff().
| . .     ef ∈ A.eff
| . .     ⇒
| . .     ef.kind = kind(a)
| . .     ⇒
| . .     x'.ef.smt.lbl = [[ef.smt.term]] 1of(e) s value(a) niltrace()
| . . 18. ∀fr:frame().
| . .     fr ∈ A.frame ⇒ ¬(kind(a) ∈ fr.acts) ⇒ x'.fr.var = s.fr.var
| . . 19. ∀x:Label. rel_mentions(r;x) ⇒ covers_var(A;x)
```

```
| . . 20. value(a) ∈ [[dec_lookup(A.da;kind(a))]] rho
| . . 21. ∀r@0:rel().
| . .     r@0 ∈ col_subst2(λx.
| . .     smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . .     ⇒
| . .     [[r@0]] rho A.ds dec_lookup(A.da;kind(a)) de e s value(a) tr
| . . 22. r' : rel()
| . . 23. r' ∈ col_subst2(λx.
| . .     smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| . . 24. tc(r';A.ds;dec_lookup(A.da;kind(a));de)
| . . 25. trace_consistent_rel(rho;A.da;tr.proj;r')
| . . 26. [[r']] rho A.ds dec_lookup(A.da;kind(a)) de e s value(a) tr
| . .     ⇒
| . .     rel_mng_2(r;
| . .               rho;
| . .               A.ds;
| . .               dec_lookup(A.da;kind(a));
| . .               de;
| . .               e;
| . .               s;
| . .               x';
| . .               value(a);
| . .               tr)
| . . 27. [[r']] rho A.ds dec_lookup(A.da;kind(a)) de e s value(a) tr
| . . ⊢  rel_mng_2(r;
| . .              rho;
| . .              A.ds;
| . .              dec_lookup(A.da;kind(a));
| . .              de;
| . .              e;
| . .              s;
| . .              x';
| . .              value(a);
| . .              tr)  by ThinTrivial THEN NthHypSq (-1) THEN Easy
|
|\
| 1. A : ioa{i:l}()
| 2. r : rel()
| 3. rho : Decl
| 4. de : sig()
| 5. e : {[[de]] rho}
| 6. a : (Σ[[A.da]] rho)
| 7. tr : trace_env([[A.da]] rho)
| 8. tc_ioa(A;de)
| 9. ¬ioa_mentions_trace(A)
| 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| 11. single_valued_decls(A.ds)
| 12. s : [[A]] rho de e.state
| 13. x' : [[A]] rho de e.state
| 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| 15. covers_rel(A;r)
| 16. [[A]] rho de e.trans(s,a,x')
| ⊢  trace_consistent_pred(rho;A.da;tr.proj;wp2_rel(A;kind(a);r))
| | by Unfold `trace_consistent_pred` 0 THEN
|
```

```
| |    Unfolds ``col_all wp2_rel`` 0 THEN Auto
|
| |    THEN  Try (Fold `pred` 0 THEN Auto)
| |
| 1. A : ioa{i:l}()
| 2. r : rel()
| 3. rho : Decl
| 4. de : sig()
| 5. e : {[[de]] rho}
| 6. a : (Σ[[A.da]] rho)
| 7. tr : trace_env([[A.da]] rho)
| 8. tc_ioa(A;de)
| 9. ¬ioa_mentions_trace(A)
| 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| 11. single_valued_decls(A.ds)
| 12. s : [[A]] rho de e.state
| 13. x' : [[A]] rho de e.state
| 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| 15. covers_rel(A;r)
| 16. [[A]] rho de e.trans(s,a,x')
| 17. r@0 : rel()
| 18. r@0 ∈ col_subst2(λx.
| |   smts_eff(action_effect(kind(a);A.eff;A.frame);x);r)
| ⊢  trace_consistent_rel(rho;A.da;tr.proj;r@0)
| | by RW0 "member_col_subst2" (-1)
| THENA Auto
|
| |    THEN ExRepD THEN HypSubstSq (-1) 0
|
| |    THEN BackThruLemma `trace_consistent_rel_subst2` THEN Auto
| |
| 1. A : ioa{i:l}()
| 2. r : rel()
| 3. rho : Decl
| 4. de : sig()
| 5. e : {[[de]] rho}
| 6. a : (Σ[[A.da]] rho)
| 7. tr : trace_env([[A.da]] rho)
| 8. tc_ioa(A;de)
| 9. ¬ioa_mentions_trace(A)
| 10. trace_consistent_rel(rho;A.da;tr.proj;r)
| 11. single_valued_decls(A.ds)
| 12. s : [[A]] rho de e.state
| 13. x' : [[A]] rho de e.state
| 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| 15. covers_rel(A;r)
| 16. [[A]] rho de e.trans(s,a,x')
| 17. r@0 : rel()
| 18. as : (Label×Term) List
| 19. 1of(unzip(as)) = rel_primed_vars(r)
| 20. ∀i:ℕ.
| |   i<||as||
| |   ⇒
| |   2of(as[i]) ∈ (λx.
| |                 smts_eff(action_effect(kind(a);A.eff;A.frame);x))
```

```
| |                (1of(as[i]))
| 21. r@0 = rel_subst2(as;r)
| ⊢  ¬subst_mentions_trace(as)
| | by AllHyps (\h. Unfold `ioa_mentions_trace` h THEN Parallel0p h)
|
| |    THEN RepeatFor 2 (OrLeft THENA Auto)
| |
| 1. A : ioa{i:l}()
| 2. r : rel()
| 3. rho : Decl
| 4. de : sig()
| 5. e : {[[de]] rho}
| 6. a : (Σ[[A.da]] rho)
| 7. tr : trace_env([[A.da]] rho)
| 8. tc_ioa(A;de)
| 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| 10. single_valued_decls(A.ds)
| 11. s : [[A]] rho de e.state
| 12. x' : [[A]] rho de e.state
| 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| 14. covers_rel(A;r)
| 15. [[A]] rho de e.trans(s,a,x')
| 16. r@0 : rel()
| 17. as : (Label×Term) List
| 18. 1of(unzip(as)) = rel_primed_vars(r)
| 19. ∀i:ℕ.
| |   i<||as||
| |   ⇒
| |   2of(as[i]) ∈ (λx.
| |                 smts_eff(action_effect(kind(a);A.eff;A.frame);x))
| |                (1of(as[i]))
| 20. r@0 = rel_subst2(as;r)
| 21. subst_mentions_trace(as)
| ⊢  ∃e:eff(). e ∈ A.eff & mentions_trace(e.smt.term)
| | by RW0 "assert_subst_mentions_trace" (-1) THENA Auto THEN ExRepD
| |
| 1. A : ioa{i:l}()
| 2. r : rel()
| 3. rho : Decl
| 4. de : sig()
| 5. e : {[[de]] rho}
| 6. a : (Σ[[A.da]] rho)
| 7. tr : trace_env([[A.da]] rho)
| 8. tc_ioa(A;de)
| 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| 10. single_valued_decls(A.ds)
| 11. s : [[A]] rho de e.state
| 12. x' : [[A]] rho de e.state
| 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| 14. covers_rel(A;r)
| 15. [[A]] rho de e.trans(s,a,x')
| 16. r@0 : rel()
| 17. as : (Label×Term) List
| 18. 1of(unzip(as)) = rel_primed_vars(r)
| 19. ∀i:ℕ.
```

```
| |   i<||as||
| |   ⇒
| |   2of(as[i]) ∈ (λx.
| |                 smts_eff(action_effect(kind(a);A.eff;A.frame);x))
| |               (1of(as[i]))
| 20. r@0 = rel_subst2(as;r)
| 21. i : ℕ||as||
| 22. mentions_trace(2of(as[i]))
| ⊢  ∃e:eff(). e ∈ A.eff & mentions_trace(e.smt.term)
| | by InstHyp [i] (-4) THENA Auto THEN Reduce (-1)
| |
| 1. A : ioa{i:l}()
| 2. r : rel()
| 3. rho : Decl
| 4. de : sig()
| 5. e : {[[de]] rho}
| 6. a : (Σ[[A.da]] rho)
| 7. tr : trace_env([[A.da]] rho)
| 8. tc_ioa(A;de)
| 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| 10. single_valued_decls(A.ds)
| 11. s : [[A]] rho de e.state
| 12. x' : [[A]] rho de e.state
| 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| 14. covers_rel(A;r)
| 15. [[A]] rho de e.trans(s,a,x')
| 16. r@0 : rel()
| 17. as : (Label×Term) List
| 18. 1of(unzip(as)) = rel_primed_vars(r)
| 19. ∀i:ℕ.
| |   i<||as||
| |   ⇒
| |   2of(as[i]) ∈ (λx.
| |                 smts_eff(action_effect(kind(a);A.eff;A.frame);x))
| |               (1of(as[i]))
| 20. r@0 = rel_subst2(as;r)
| 21. i : ℕ||as||
| 22. mentions_trace(2of(as[i]))
| 23. 2of(as[i]) ∈ smts_eff(action_effect(kind(a);A.eff;A.frame);...)
| ⊢  ∃e:eff(). e ∈ A.eff & mentions_trace(e.smt.term)
| | by Unfold `smts_eff` (-1) THEN Unfold `smt_terms` (-1)
|
| |    THEN RW ColMemberC (-1) THEN ExRepD
|
| |    THEN Unfold `action_effect` (-3)
| THEN RW ColMemberC (-3)
|
| |    THEN D (-3) THEN ExRepD
| |
| |\
| | 1. A : ioa{i:l}()
| | 2. r : rel()
| | 3. rho : Decl
| | 4. de : sig()
| | 5. e : {[[de]] rho}
```

```
| | 6. a : (Σ[[A.da]] rho)
| | 7. tr : trace_env([[A.da]] rho)
| | 8. tc_ioa(A;de)
| | 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| | 10. single_valued_decls(A.ds)
| | 11. s : [[A]] rho de e.state
| | 12. x' : [[A]] rho de e.state
| | 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| | 14. covers_rel(A;r)
| | 15. [[A]] rho de e.trans(s,a,x')
| | 16. r@0 : rel()
| | 17. as : (Label×Term) List
| | 18. 1of(unzip(as)) = rel_primed_vars(r)
| | 19. ∀i:ℕ.
| |     i<||as||
| |     ⇒
| |     2of(as[i]) ∈ ...
| |                  (1of(as[i]))
| | 20. r@0 = rel_subst2(as;r)
| | 21. i : ℕ||as||
| | 22. mentions_trace(2of(as[i]))
| | 23. s1 : smt()
| | 24. e1 : eff()
| | 25. e1 ∈ A.eff
| | 26. e1.kind =1 kind(a)
| | 27. s1 = e1.smt
| | 28. s1.lbl =1 1of(as[i])
| | 29. 2of(as[i]) = s1.term
| | ⊢  ∃e:eff(). e ∈ A.eff & mentions_trace(e.smt.term)
| |   by InstConcl [e1] THEN Auto
| | THEN SubstFor e1.smt 0
| |
| |      THENM SubstFor s1.term 0
| | THEN Auto
| |
| .\
| . 1. A : ioa{i:l}()
| . 2. r : rel()
| . 3. rho : Decl
| . 4. de : sig()
| . 5. e : {[[de]] rho}
| . 6. a : (Σ[[A.da]] rho)
| . 7. tr : trace_env([[A.da]] rho)
| . 8. tc_ioa(A;de)
| . 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| . 10. single_valued_decls(A.ds)
| . 11. s : [[A]] rho de e.state
| . 12. x' : [[A]] rho de e.state
| . 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . 14. covers_rel(A;r)
| . 15. [[A]] rho de e.trans(s,a,x')
| . 16. r@0 : rel()
| . 17. as : (Label×Term) List
| . 18. 1of(unzip(as)) = rel_primed_vars(r)
| . 19. ∀i:ℕ.
```

```
| . |   i<||as||
| . |   ⇒
| . |   2of(as[i]) ∈ ...
| . |                (1of(as[i]))
| . 20. r@0 = rel_subst2(as;r)
| . 21. i : ℕ||as||
| . 22. mentions_trace(2of(as[i]))
| . 23. s1 : smt()
| . 24. f : frame()
| . 25. f ∈ A.frame
| . 26. ¬₂kind(a) ∈ f.acts
| . 27. s1 = mk_smt(f.var, f.var, f.typ)
| . 28. s1.lbl =₁ 1of(as[i])
| . 29. 2of(as[i]) = s1.term
| . ⊢  ∃e:eff(). e ∈ A.eff & mentions_trace(e.smt.term)
| . | by Assert ¬mentions_trace(2of(as[i])) THEN Auto
| . THEN SubstFor
| . |    2of(as[i]) 0
| . THENM SubstFor s1 0 THEN Auto
| . |
| . 1. A : ioa{i:l}()
| . 2. r : rel()
| . 3. rho : Decl
| . 4. de : sig()
| . 5. e : {[[de]] rho}
| . 6. a : (Σ[[A.da]] rho)
| . 7. tr : trace_env([[A.da]] rho)
| . 8. tc_ioa(A;de)
| . 9. trace_consistent_rel(rho;A.da;tr.proj;r)
| . 10. single_valued_decls(A.ds)
| . 11. s : [[A]] rho de e.state
| . 12. x' : [[A]] rho de e.state
| . 13. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
| . 14. covers_rel(A;r)
| . 15. [[A]] rho de e.trans(s,a,x')
| . 16. r@0 : rel()
| . 17. as : (Label×Term) List
| . 18. 1of(unzip(as)) = rel_primed_vars(r)
| . 19. ∀i:ℕ.
| .     i<||as||
| .     ⇒
| .     2of(as[i]) ∈ ...
| .                  (1of(as[i]))
| . 20. r@0 = rel_subst2(as;r)
| . 21. i : ℕ||as||
| . 22. mentions_trace(2of(as[i]))
| . 23. s1 : smt()
| . 24. f : frame()
| . 25. f ∈ A.frame
| . 26. ¬₂kind(a) ∈ f.acts
| . 27. s1 = mk_smt(f.var, f.var, f.typ)
| . 28. s1.lbl =₁ 1of(as[i])
| . 29. 2of(as[i]) = s1.term
| . ⊢  ¬mentions_trace(mk_smt(f.var, f.var, f.typ).term)
| .   by Reduce 0 THEN SimpConcl
```

```
|
.\
. 1. A : ioa{i:l}()
. 2. r : rel()
. 3. rho : Decl
. 4. de : sig()
. 5. e : {[[de]] rho}
. 6. a : (Σ[[A.da]] rho)
. 7. tr : trace_env([[A.da]] rho)
. 8. tc_ioa(A;de)
. 9. ¬ioa_mentions_trace(A)
. 10. trace_consistent_rel(rho;A.da;tr.proj;r)
. 11. single_valued_decls(A.ds)
. 12. s : [[A]] rho de e.state
. 13. x' : [[A]] rho de e.state
. 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
. 15. covers_rel(A;r)
. 16. [[A]] rho de e.trans(s,a,x')
. ⊢  tc_pred(wp2_rel(A;kind(a);r);A.ds;dec_lookup(A.da;kind(a));de)
. | by InstLemma `tc_wp2` [A;r;de;kind(a)] THENA Auto
. |
. |\
. | .....antecedent .....
. | 1. A : ioa{i:l}()
. | 2. r : rel()
. | 3. rho : Decl
. | 4. de : sig()
. | 5. e : {[[de]] rho}
. | 6. a : (Σ[[A.da]] rho)
. | 7. tr : trace_env([[A.da]] rho)
. | 8. tc_ioa(A;de)
. | 9. ¬ioa_mentions_trace(A)
. | 10. trace_consistent_rel(rho;A.da;tr.proj;r)
. | 11. single_valued_decls(A.ds)
. | 12. s : [[A]] rho de e.state
. | 13. x' : [[A]] rho de e.state
. | 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
. | 15. covers_rel(A;r)
. | 16. [[A]] rho de e.trans(s,a,x')
. | ⊢  tc_pred(r;A.ds;dec_lookup(A.da;kind(a));de)
. |   by Unfolds ``tc_pred pred_rel`` 0 THEN RW ColMemberC 0 THEN Auto
. |
. |   THEN Subst r@0 ∼ r 0  THEN Auto
. |
. .\
. . 1. A : ioa{i:l}()
. . 2. r : rel()
. . 3. rho : Decl
. . 4. de : sig()
. . 5. e : {[[de]] rho}
. . 6. a : (Σ[[A.da]] rho)
. . 7. tr : trace_env([[A.da]] rho)
. . 8. tc_ioa(A;de)
. . 9. ¬ioa_mentions_trace(A)
. . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
```

```
. . 11. single_valued_decls(A.ds)
. . 12. s : [[A]] rho de e.state
. . 13. x' : [[A]] rho de e.state
. . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
. . 15. covers_rel(A;r)
. . 16. [[A]] rho de e.trans(s,a,x')
. . 17. tc_pred(wp2(A;kind(a);r);A.ds;dec_lookup(A.da;kind(a));de)
. . ⊢  tc_pred(wp2_rel(A;kind(a);r);A.ds;dec_lookup(A.da;kind(a));de)
. . | by Unfolds ``wp2 pred_rel`` (-1) THEN Fold `wp2_rel` (-1)
. .
. . |       THEN All (Unfold `tc_pred`) THEN
. .
. . |    ...
. . | THEN BackThruSomeHyp THEN Auto
. . |
. . 1. A : ioa{i:l}()
. . 2. r : rel()
. . 3. rho : Decl
. . 4. de : sig()
. . 5. e : {[[de]] rho}
. . 6. a : (Σ[[A.da]] rho)
. . 7. tr : trace_env([[A.da]] rho)
. . 8. tc_ioa(A;de)
. . 9. ¬ioa_mentions_trace(A)
. . 10. trace_consistent_rel(rho;A.da;tr.proj;r)
. . 11. single_valued_decls(A.ds)
. . 12. s : [[A]] rho de e.state
. . 13. x' : [[A]] rho de e.state
. . 14. tc(r;A.ds;dec_lookup(A.da;kind(a));de)
. . 15. covers_rel(A;r)
. . 16. [[A]] rho de e.trans(s,a,x')
. . 17. ∀r@0:rel().
. .     (∃r@1:rel(). r@1 = r & r@0 ∈ wp2_rel(A;kind(a);r@1))
. .     ⇒
. .     tc(r@0;A.ds;dec_lookup(A.da;kind(a));de)
. . 18. r@0 : rel()
. . 19. r@0 ∈ wp2_rel(A;kind(a);r)
. . ⊢  ∃r@1:rel(). r@1 = r & r@0 ∈ wp2_rel(A;kind(a);r@1)
. .   by InstConcl [r] THEN Auto THEN Try (Fold `pred` 0 THEN Auto)
END
```