```
vc_trace_correct_action_decl_lemma  THM

⊢  ∀A:ioa{i:l}(), I:Fmla, rho:Decl, de:sig(), e:{[[de]] rho}
|  , te:(Label→Label→𝔹).
|  tc_ioa(A;de)
|  ⇒
|  ¬ioa_mentions_trace(A)
|  ⇒
|  trace_consistent_pred(rho;A.da;te;I)
|  ⇒
|  tc_pred(I;A.ds;<>;de)
|  ⇒
|  covers_pred(A;I)
|  ⇒
|  guarded_trace(A.da;te;I)
|  ⇒
|  closed_pred(I)
|  ⇒
|  single_valued_decls(A.ds)
|  ⇒
|  (∀s0,x:[[A]] rho de e.state, act:[[A]] rho de e.action
|   , x':[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
|   [[A]] rho de e.init(s0)
|   ⇒
|   trace_reachable([[A]] rho de e;s0;mk_trace_env(tr, te).trace;x)
|   ⇒
|   [[I]] rho A.ds <> de e x · mk_trace_env(tr, te)
|   ⇒
|   [[A]] rho de e.trans(x,act,x')
|   ⇒
|   ((∃t:dec(). t ∈ A.da & t.lbl = kind(act))
|    ⇒
|    [[I]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act))
|   ⇒
|   [[I]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act))
| by let T1 = Unfold `ioa_mng` 0 THEN Reduce 0 THEN Trivial in

|    ...
|  let T = Auto THEN Try (Unfold `decl` 0 THEN Auto)

|                   THEN Try (Fold `pred` 0 THEN Auto)

|    ...
|               THEN TrivialIoaHyp

|                    THEN Try T1
|               THEN Try T2 in

|      RepeatFor 24 (D 0 THENA Complete T)
|
1. A : ioa{i:l}()
2. I : Fmla
3. rho : Decl
4. de : sig()
5. e : {[[de]] rho}
```

```
6. te : Label→Label→𝔹
7. tc_ioa(A;de)
8. ¬ioa_mentions_trace(A)
9. trace_consistent_pred(rho;A.da;te;I)
10. tc_pred(I;A.ds;<>;de)
11. covers_pred(A;I)
12. guarded_trace(A.da;te;I)
13. closed_pred(I)
14. single_valued_decls(A.ds)
15. s0 : [[A]] rho de e.state
16. x : [[A]] rho de e.state
17. act : [[A]] rho de e.action
18. x' : [[A]] rho de e.state
19. tr : (Σ[[A.da]] rho) List
20. [[A]] rho de e.init(s0)
21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr, te).trace;x)
22. [[I]] rho A.ds <> de e x · mk_trace_env(tr, te)
23. [[A]] rho de e.trans(x,act,x')
24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
|   ⇒
|   [[I]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act)
⊢  [[I]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act)
| by  Assert
|    [[wp(A;kind(act);I)]] rho A.ds dec_lookup(...;...) de e x ... ...
|    ⇒
|    [[I]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act)
|
|\
| .....assertion .....
| 1. A : ioa{i:l}()
| 2. I : Fmla
| 3. rho : Decl
| 4. de : sig()
| 5. e : {[[de]] rho}
| 6. te : Label→Label→𝔹
| 7. tc_ioa(A;de)
| 8. ¬ioa_mentions_trace(A)
| 9. trace_consistent_pred(rho;A.da;te;I)
| 10. tc_pred(I;A.ds;<>;de)
| 11. covers_pred(A;I)
| 12. guarded_trace(A.da;te;I)
| 13. closed_pred(I)
| 14. single_valued_decls(A.ds)
| 15. s0 : [[A]] rho de e.state
| 16. x : [[A]] rho de e.state
| 17. act : [[A]] rho de e.action
| 18. x' : [[A]] rho de e.state
| 19. tr : (Σ[[A.da]] rho) List
| 20. [[A]] rho de e.init(s0)
| 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr, te).trace;x)
| 22. [[I]] rho A.ds <> de e x · mk_trace_env(tr, te)
| 23. [[A]] rho de e.trans(x,act,x')
| 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
| |   ⇒
| |   [[I]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act)
```

```
| ⊢  [[wp(A;kind(act);I)]] rho A.ds dec_lookup(...;...) de e x ... ...
| |  ⇒
| |  [[I]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act)
| | by Assert act ∈ (Σ[[A.da]] rho)
| |
| |\
| | .....assertion .....
| | 1. A : ioa{i:l}()
| | 2. I : Fmla
| | 3. rho : Decl
| | 4. de : sig()
| | 5. e : {[[de]] rho}
| | 6. te : Label→Label→𝔹
| | 7. tc_ioa(A;de)
| | 8. ¬ioa_mentions_trace(A)
| | 9. trace_consistent_pred(rho;A.da;te;I)
| | 10. tc_pred(I;A.ds;<>;de)
| | 11. covers_pred(A;I)
| | 12. guarded_trace(A.da;te;I)
| | 13. closed_pred(I)
| | 14. single_valued_decls(A.ds)
| | 15. s0 : [[A]] rho de e.state
| | 16. x : [[A]] rho de e.state
| | 17. act : [[A]] rho de e.action
| | 18. x' : [[A]] rho de e.state
| | 19. tr : (Σ[[A.da]] rho) List
| | 20. [[A]] rho de e.init(s0)
| | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
| |                                     te).trace;x)
| | 22. [[I]] rho A.ds <> de e x · mk_trace_env(tr, te)
| | 23. [[A]] rho de e.trans(x,act,x')
| | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
| |     ⇒
| |     [[I]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act)
| | ⊢  act ∈ (Σ[[A.da]] rho)  by TrivialIoaHyp
| |
| .\
| . 1. A : ioa{i:l}()
| . 2. I : Fmla
| . 3. rho : Decl
| . 4. de : sig()
| . 5. e : {[[de]] rho}
| . 6. te : Label→Label→𝔹
| . 7. tc_ioa(A;de)
| . 8. ¬ioa_mentions_trace(A)
| . 9. trace_consistent_pred(rho;A.da;te;I)
| . 10. tc_pred(I;A.ds;<>;de)
| . 11. covers_pred(A;I)
| . 12. guarded_trace(A.da;te;I)
| . 13. closed_pred(I)
| . 14. single_valued_decls(A.ds)
| . 15. s0 : [[A]] rho de e.state
| . 16. x : [[A]] rho de e.state
| . 17. act : [[A]] rho de e.action
| . 18. x' : [[A]] rho de e.state
```

```
| . 19. tr : (Σ[[A.da]] rho) List
| . 20. [[A]] rho de e.init(s0)
| . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
| .                                    te).trace;x)
| . 22. [[I]] rho A.ds <> de e x · mk_trace_env(tr, te)
| . 23. [[A]] rho de e.trans(x,act,x')
| . 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
| .     ⇒
| .     [[I]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act)
| . 25. act ∈ (Σ[[A.da]] rho)
| . ⊢  [[wp(A;kind(act);I)]] rho A.ds ... de e x value(act) ...
| .    ⇒
| .    [[I]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act)
| .   by InstLemma `wp_correctness` [A;de;I;rho;e;act;
| .      tappend(mk_trace_env(tr, te);act);x;x']
| .
| .      ...
| . THEN D (-1) THEN Try Trivial
|
.\
. 1. A : ioa{i:l}()
. 2. I : Fmla
. 3. rho : Decl
. 4. de : sig()
. 5. e : {[[de]] rho}
. 6. te : Label→Label→𝔹
. 7. tc_ioa(A;de)
. 8. ¬ioa_mentions_trace(A)
. 9. trace_consistent_pred(rho;A.da;te;I)
. 10. tc_pred(I;A.ds;<>;de)
. 11. covers_pred(A;I)
. 12. guarded_trace(A.da;te;I)
. 13. closed_pred(I)
. 14. single_valued_decls(A.ds)
. 15. s0 : [[A]] rho de e.state
. 16. x : [[A]] rho de e.state
. 17. act : [[A]] rho de e.action
. 18. x' : [[A]] rho de e.state
. 19. tr : (Σ[[A.da]] rho) List
. 20. [[A]] rho de e.init(s0)
. 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr, te).trace;x)
. 22. [[I]] rho A.ds <> de e x · mk_trace_env(tr, te)
. 23. [[A]] rho de e.trans(x,act,x')
. 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. |   ⇒
. |   [[I]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act)
. 25. [[wp(A;kind(act);I)]] rho A.ds ... de e x value(act) ...
. |   ⇒
. |   [[I]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act)
. ⊢  [[I]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act)
. | by All (Unfolds ``pred pred_mng``) THEN UnivCD THENA Auto
. |
. 1. A : ioa{i:l}()
. 2. I : Collection(rel())
. 3. rho : Decl
```

```
. 4. de : sig()
. 5. e : {[[de]] rho}
. 6. te : Label→Label→𝔹
. 7. tc_ioa(A;de)
. 8. ¬ioa_mentions_trace(A)
. 9. trace_consistent_pred(rho;A.da;te;I)
. 10. tc_pred(I;A.ds;<>;de)
. 11. covers_pred(A;I)
. 12. guarded_trace(A.da;te;I)
. 13. closed_pred(I)
. 14. single_valued_decls(A.ds)
. 15. s0 : [[A]] rho de e.state
. 16. x : [[A]] rho de e.state
. 17. act : [[A]] rho de e.action
. 18. x' : [[A]] rho de e.state
. 19. tr : (Σ[[A.da]] rho) List
. 20. [[A]] rho de e.init(s0)
. 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr, te).trace;x)
. 22. ∀r:rel().
. |   r ∈ I ⇒ [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. 23. [[A]] rho de e.trans(x,act,x')
. 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. |   ⇒
. |   (∀r:rel().
. |    r ∈ I
. |    ⇒
. |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act))
. 25. ...
. |   ⇒
. |   (∀r:rel().
. |    r ∈ I
. |    ⇒
. |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act))
. 26. r : rel()
. 27. r ∈ I
. ⊢  [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act)
. | by ...
. THEN RW ColMemberC (-3)
. |
. 1. A : ioa{i:l}()
. 2. I : Collection(rel())
. 3. rho : Decl
. 4. de : sig()
. 5. e : {[[de]] rho}
. 6. te : Label→Label→𝔹
. 7. tc_ioa(A;de)
. 8. ¬ioa_mentions_trace(A)
. 9. trace_consistent_pred(rho;A.da;te;I)
. 10. tc_pred(I;A.ds;<>;de)
. 11. covers_pred(A;I)
. 12. guarded_trace(A.da;te;I)
. 13. closed_pred(I)
. 14. single_valued_decls(A.ds)
. 15. s0 : [[A]] rho de e.state
. 16. x : [[A]] rho de e.state
```

```
. 17. act : [[A]] rho de e.action
. 18. x' : [[A]] rho de e.state
. 19. tr : (Σ[[A.da]] rho) List
. 20. [[A]] rho de e.init(s0)
. 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr, te).trace;x)
. 22. ∀r:rel().
. |   r ∈ I ⇒ [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. 23. [[A]] rho de e.trans(x,act,x')
. 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. |   ⇒
. |   (∀r:rel().
. |    r ∈ I
. |    ⇒
. |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act))
. 25. ...
. |   ⇒
. |   (∀r:rel().
. |    r ∈ I
. |    ⇒
. |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act))
. 26. r : rel()
. 27. r ∈ I
. ⊢  [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act)
. | by Assert act ∈ (Σ[[A.da]] rho)
. |
. |
. |\
. | .....assertion .....
. | 1. A : ioa{i:l}()
. | 2. I : Collection(rel())
. | 3. rho : Decl
. | 4. de : sig()
. | 5. e : {[[de]] rho}
. | 6. te : Label→Label→𝔹
. | 7. tc_ioa(A;de)
. | 8. ¬ioa_mentions_trace(A)
. | 9. trace_consistent_pred(rho;A.da;te;I)
. | 10. tc_pred(I;A.ds;<>;de)
. | 11. covers_pred(A;I)
. | 12. guarded_trace(A.da;te;I)
. | 13. closed_pred(I)
. | 14. single_valued_decls(A.ds)
. | 15. s0 : [[A]] rho de e.state
. | 16. x : [[A]] rho de e.state
. | 17. act : [[A]] rho de e.action
. | 18. x' : [[A]] rho de e.state
. | 19. tr : (Σ[[A.da]] rho) List
. | 20. [[A]] rho de e.init(s0)
. | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. |                                      te).trace;x)
. | 22. ∀r:rel().
. |     r ∈ I ⇒ [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. | 23. [[A]] rho de e.trans(x,act,x')
. | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. |     ⇒
```

```
. |     (∀r:rel().
. |      r ∈ I
. |      ⇒
. |      [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. |                                              te);act))
. | 25. (∀r:rel().
. |      (∃r@0:rel().
. |       r@0 ∈ I
. |       & r ∈ smts_eff_rel(<e.smt | e ∈ <e ∈ A.eff |
. |                                           e.kind =1 kind(act)>>
. |         + <mk_smt(f.var, f.var, f.typ) |
. |            f ∈ <f ∈ A.frame | ¬2kind(act) ∈ f.acts>>;r@0))
. |      ⇒
. |      [[r]] rho A.ds dec_lookup(A.da;kind(act)) de e x ... ...)
. |     ⇒
. |     (∀r:rel().
. |      r ∈ I
. |      ⇒
. |      [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. |                                              te);act))
. | 26. r : rel()
. | 27. r ∈ I
. | ⊢  act ∈ (Σ[[A.da]] rho)  by TrivialIoaHyp
. |
. .\
. . 1. A : ioa{i:l}()
. . 2. I : Collection(rel())
. . 3. rho : Decl
. . 4. de : sig()
. . 5. e : {[[de]] rho}
. . 6. te : Label→Label→𝔹
. . 7. tc_ioa(A;de)
. . 8. ¬ioa_mentions_trace(A)
. . 9. trace_consistent_pred(rho;A.da;te;I)
. . 10. tc_pred(I;A.ds;<>;de)
. . 11. covers_pred(A;I)
. . 12. guarded_trace(A.da;te;I)
. . 13. closed_pred(I)
. . 14. single_valued_decls(A.ds)
. . 15. s0 : [[A]] rho de e.state
. . 16. x : [[A]] rho de e.state
. . 17. act : [[A]] rho de e.action
. . 18. x' : [[A]] rho de e.state
. . 19. tr : (Σ[[A.da]] rho) List
. . 20. [[A]] rho de e.init(s0)
. . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . |                                         te).trace;x)
. . 22. ∀r:rel().
. . |   r ∈ I ⇒ [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . 23. [[A]] rho de e.trans(x,act,x')
. . 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . |   ⇒
. . |   (∀r:rel().
. . |    r ∈ I
. . |    ⇒
```

```
. . |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . |                                            te);act))
. . 25. (∀r:rel().
. . |    (∃r@0:rel().
. . |     r@0 ∈ I
. . |     & r ∈ smts_eff_rel(<e.smt | e ∈ <e ∈ A.eff |
. . |                                         e.kind =₁ kind(act)>>
. . |       + <mk_smt(f.var, f.var, f.typ) |
. . |          f ∈ <f ∈ A.frame | ¬₂kind(act) ∈ f.acts>>;r@0))
. . |    ⇒
. . |    [[r]] rho A.ds dec_lookup(A.da;kind(act)) de e x ... ...)
. . |   ⇒
. . |   (∀r:rel().
. . |    r ∈ I
. . |    ⇒
. . |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . |                                            te);act))
. . 26. r : rel()
. . 27. r ∈ I
. . 28. act ∈ (Σ[[A.da]] rho)
. . ⊢  [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr, te);act)
. . | by InstLemma `wp_rel_correctness` [A;r;rho;de;e;act;
. . |    tappend(mk_trace_env(tr, te);act);x;x'] THENA Auto
. .
. . |    ...
. . |
. . |\
. . | .....antecedent .....
. . | 1. A : ioa{i:l}()
. . | 2. I : Collection(rel())
. . | 3. rho : Decl
. . | 4. de : sig()
. . | 5. e : {[[de]] rho}
. . | 6. te : Label→Label→𝔹
. . | 7. tc_ioa(A;de)
. . | 8. ¬ioa_mentions_trace(A)
. . | 9. trace_consistent_pred(rho;A.da;te;I)
. . | 10. tc_pred(I;A.ds;<>;de)
. . | 11. covers_pred(A;I)
. . | 12. guarded_trace(A.da;te;I)
. . | 13. closed_pred(I)
. . | 14. single_valued_decls(A.ds)
. . | 15. s0 : [[A]] rho de e.state
. . | 16. x : [[A]] rho de e.state
. . | 17. act : [[A]] rho de e.action
. . | 18. x' : [[A]] rho de e.state
. . | 19. tr : (Σ[[A.da]] rho) List
. . | 20. [[A]] rho de e.init(s0)
. . | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . |                                            te).trace;x)
. . | 22. ∀r:rel().
. . |     r ∈ I ⇒ [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . | 23. [[A]] rho de e.trans(x,act,x')
. . | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . |     ⇒
```

```
. . |     (∀r:rel().
. . |      r ∈ I
. . |      ⇒
. . |      [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . |                                            te);act))
. . | 25. (∀r:rel().
. . |      (∃r@0:rel().
. . |       r@0 ∈ I
. . |       & r ∈ smts_eff_rel(<e.smt | e ∈ <e ∈ A.eff |
. . |                                          e.kind =₁ kind(act)>>
. . |         + <mk_smt(f.var, f.var, f.typ) |
. . |            f ∈ <f ∈ A.frame | ¬₂kind(act) ∈ f.acts>>;r@0))
. . |      ⇒
. . |      [[r]] rho A.ds dec_lookup(A.da;kind(act)) de e x ... ...)
. . |     ⇒
. . |     (∀r:rel().
. . |      r ∈ I
. . |      ⇒
. . |      [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . |                                            te);act))
. . | 26. r : rel()
. . | 27. r ∈ I
. . | 28. act ∈ (Σ[[A.da]] rho)
. . | ⊢  covers_rel(A;r)  by Easy
. . |
. . .\
. . . 1. A : ioa{i:l}()
. . . 2. I : Collection(rel())
. . . 3. rho : Decl
. . . 4. de : sig()
. . . 5. e : {[[de]] rho}
. . . 6. te : Label→Label→𝔹
. . . 7. tc_ioa(A;de)
. . . 8. ¬ioa_mentions_trace(A)
. . . 9. trace_consistent_pred(rho;A.da;te;I)
. . . 10. tc_pred(I;A.ds;<>;de)
. . . 11. covers_pred(A;I)
. . . 12. guarded_trace(A.da;te;I)
. . . 13. closed_pred(I)
. . . 14. single_valued_decls(A.ds)
. . . 15. s0 : [[A]] rho de e.state
. . . 16. x : [[A]] rho de e.state
. . . 17. act : [[A]] rho de e.action
. . . 18. x' : [[A]] rho de e.state
. . . 19. tr : (Σ[[A.da]] rho) List
. . . 20. [[A]] rho de e.init(s0)
. . . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . |                                      te).trace;x)
. . . 22. ∀r:rel().
. . . |   r ∈ I ⇒ [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . 23. [[A]] rho de e.trans(x,act,x')
. . . 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . |   ⇒
. . . |   (∀r:rel().
. . . |    r ∈ I
```

```
. . . |     ⇒
. . . |     [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . |                                          te);act))
. . . 25. (∀r:rel().
. . . |     (∃r@0:rel().
. . . |      r@0 ∈ I
. . . |      & r ∈ smts_eff_rel(<e.smt | e ∈ <e ∈ A.eff |
. . . |                                        e.kind =1 kind(act)>>
. . . |        + <mk_smt(f.var, f.var, f.typ) |
. . . |           f ∈ <f ∈ A.frame | ¬2kind(act) ∈ f.acts>>;r@0))
. . . |     ⇒
. . . |     [[r]] rho A.ds dec_lookup(A.da;kind(act)) de e x ... ...)
. . . |    ⇒
. . . |    (∀r:rel().
. . . |     r ∈ I
. . . |     ⇒
. . . |     [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . |                                          te);act))
. . . 26. r : rel()
. . . 27. r ∈ I
. . . 28. act ∈ (Σ[[A.da]] rho)
. . . 29. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . |                                        te);act)
. . . |    ⇔
. . . |    [[wp_rel(A;kind(act);r)]] rho A.ds ... de e x value(act) ...
. . . ⊢  [[r]] rho A.ds <> de e x' · mk_trace_env(tr @ [act], te)
. . . | by D (-1) THEN D (-1)
. . . |
. . . |\
. . . | .....antecedent .....
. . . | 1. A : ioa{i:l}()
. . . | 2. I : Collection(rel())
. . . | 3. rho : Decl
. . . | 4. de : sig()
. . . | 5. e : {[[de]] rho}
. . . | 6. te : Label→Label→𝔹
. . . | 7. tc_ioa(A;de)
. . . | 8. ¬ioa_mentions_trace(A)
. . . | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | 10. tc_pred(I;A.ds;<>;de)
. . . | 11. covers_pred(A;I)
. . . | 12. guarded_trace(A.da;te;I)
. . . | 13. closed_pred(I)
. . . | 14. single_valued_decls(A.ds)
. . . | 15. s0 : [[A]] rho de e.state
. . . | 16. x : [[A]] rho de e.state
. . . | 17. act : [[A]] rho de e.action
. . . | 18. x' : [[A]] rho de e.state
. . . | 19. tr : (Σ[[A.da]] rho) List
. . . | 20. [[A]] rho de e.init(s0)
. . . | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | |                                       te).trace;x)
. . . | 22. ∀r:rel().
. . . | |    r ∈ I ⇒ [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | 23. [[A]] rho de e.trans(x,act,x')
```

```
. . . | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | |   ⇒
. . . | |   (∀r:rel().
. . . | |    r ∈ I
. . . | |    ⇒
. . . | |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | |                                           te);act))
. . . | 25. (∀r:rel().
. . . | |    (∃r@0:rel().
. . . | |     r@0 ∈ I
. . . | |     & r ∈ smts_eff_rel(<e.smt | e ∈ <e ∈ A.eff |
. . . | |                                       e.kind =₁ kind(act)>>
. . . | |        + <mk_smt(f.var, f.var, f.typ) |
. . . | |           f ∈ <f ∈ A.frame | ¬₂kind(act) ∈ f.acts>>;r@0))
. . . | |    ⇒
. . . | |    [[r]] rho A.ds dec_lookup(A.da;kind(act)) de e x ... ...)
. . . | |   ⇒
. . . | |   (∀r:rel().
. . . | |    r ∈ I
. . . | |    ⇒
. . . | |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | |                                           te);act))
. . . | 26. r : rel()
. . . | 27. r ∈ I
. . . | 28. act ∈ (Σ[[A.da]] rho)
. . . | 29. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | |                                         te);act)
. . . | |   ⇒
. . . | |   [[wp_rel(A;kind(act);r)]] rho A.ds ... de e x ... ...
. . . | ⊢  [[wp_rel(A;kind(act);r)]] rho A.ds ... de e x ... ...
. . . | | by ...
. . . | |
. . . | 1. A : ioa{i:l}()
. . . | 2. I : Collection(rel())
. . . | 3. rho : Decl
. . . | 4. de : sig()
. . . | 5. e : {[[de]] rho}
. . . | 6. te : Label→Label→𝔹
. . . | 7. tc_ioa(A;de)
. . . | 8. ¬ioa_mentions_trace(A)
. . . | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | 10. tc_pred(I;A.ds;<>;de)
. . . | 11. covers_pred(A;I)
. . . | 12. guarded_trace(A.da;te;I)
. . . | 13. closed_pred(I)
. . . | 14. single_valued_decls(A.ds)
. . . | 15. s0 : [[A]] rho de e.state
. . . | 16. x : [[A]] rho de e.state
. . . | 17. act : [[A]] rho de e.action
. . . | 18. x' : [[A]] rho de e.state
. . . | 19. tr : (Σ[[A.da]] rho) List
. . . | 20. [[A]] rho de e.init(s0)
. . . | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | |                                    te).trace;x)
. . . | 22. ∀r:rel().
```

```
. . . | |   r ∈ I ⇒ [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | 23. [[A]] rho de e.trans(x,act,x')
. . . | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | |   ⇒
. . . | |   (∀r:rel().
. . . | |    r ∈ I
. . . | |    ⇒
. . . | |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | |                                        te);act))
. . . | 25. (∀r:rel().
. . . | |    (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | |    ⇒
. . . | |    [[r]] rho A.ds dec_lookup(A.da;kind(act)) de e x ... ...)
. . . | |   ⇒
. . . | |   (∀r:rel().
. . . | |    r ∈ I
. . . | |    ⇒
. . . | |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | |                                        te);act))
. . . | 26. r : rel()
. . . | 27. r ∈ I
. . . | 28. act ∈ (Σ[[A.da]] rho)
. . . | 29. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | |                                        te);act)
. . . | |   ⇒
. . . | |   [[wp_rel(A;kind(act);r)]] rho A.ds ... de e x ... ...
. . . | ⊢  [[wp_rel(A;kind(act);r)]] rho A.ds ... de e x ... ...
. . . | | by ...
. . . | |
. . . | 1. A : ioa{i:l}()
. . . | 2. I : Collection(rel())
. . . | 3. rho : Decl
. . . | 4. de : sig()
. . . | 5. e : {[[de]] rho}
. . . | 6. te : Label→Label→𝔹
. . . | 7. tc_ioa(A;de)
. . . | 8. ¬ioa_mentions_trace(A)
. . . | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | 10. tc_pred(I;A.ds;<>;de)
. . . | 11. covers_pred(A;I)
. . . | 12. guarded_trace(A.da;te;I)
. . . | 13. closed_pred(I)
. . . | 14. single_valued_decls(A.ds)
. . . | 15. s0 : [[A]] rho de e.state
. . . | 16. x : [[A]] rho de e.state
. . . | 17. act : [[A]] rho de e.action
. . . | 18. x' : [[A]] rho de e.state
. . . | 19. tr : (Σ[[A.da]] rho) List
. . . | 20. [[A]] rho de e.init(s0)
. . . | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | |                                      te).trace;x)
. . . | 22. ∀r:rel().
. . . | |   r ∈ I ⇒ [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | 23. [[A]] rho de e.trans(x,act,x')
. . . | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
```

```
. . . | |    ⇒
. . . | |    (∀r:rel().
. . . | |     r ∈ I
. . . | |     ⇒
. . . | |     [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | |                                          te);act))
. . . | 25. (∀r:rel().
. . . | |     (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | |     ⇒
. . . | |     [[r]] rho A.ds dec_lookup(A.da;kind(act)) de e x ... ...)
. . . | |    ⇒
. . . | |    (∀r:rel().
. . . | |     r ∈ I
. . . | |     ⇒
. . . | |     [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | |                                          te);act))
. . . | 26. r : rel()
. . . | 27. r ∈ I
. . . | 28. act ∈ (Σ[[A.da]] rho)
. . . | 29. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | |                                         te);act)
. . . | |    ⇒
. . . | |    [[wp_rel(A;kind(act);r)]] rho A.ds ... de e x ... ...
. . . | 30. r@0 : rel()
. . . | 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | ⊢  [[r@0]] rho A.ds dec_lookup(A.da;kind(act)) de e x ... ...
. . . | | by Decide rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | |     THENA Auto
. . . | |
. . . | |\
. . . | | 1. A : ioa{i:l}()
. . . | | 2. I : Collection(rel())
. . . | | 3. rho : Decl
. . . | | 4. de : sig()
. . . | | 5. e : {[[de]] rho}
. . . | | 6. te : Label→Label→𝔹
. . . | | 7. tc_ioa(A;de)
. . . | | 8. ¬ioa_mentions_trace(A)
. . . | | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | 10. tc_pred(I;A.ds;<>;de)
. . . | | 11. covers_pred(A;I)
. . . | | 12. guarded_trace(A.da;te;I)
. . . | | 13. closed_pred(I)
. . . | | 14. single_valued_decls(A.ds)
. . . | | 15. s0 : [[A]] rho de e.state
. . . | | 16. x : [[A]] rho de e.state
. . . | | 17. act : [[A]] rho de e.action
. . . | | 18. x' : [[A]] rho de e.state
. . . | | 19. tr : (Σ[[A.da]] rho) List
. . . | | 20. [[A]] rho de e.init(s0)
. . . | | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | | |                                         te).trace;x)
. . . | | 22. ∀r:rel().
. . . | | |    r ∈ I ⇒ [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | 23. [[A]] rho de e.trans(x,act,x')
```

```
. . . | | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | |   ⇒
. . . | | |   (∀r:rel().
. . . | | |    r ∈ I
. . . | | |    ⇒
. . . | | |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | |                                         te);act))
. . . | | 25. (∀r:rel().
. . . | | |    (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | |    ⇒
. . . | | |    [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...)
. . . | | |   ⇒
. . . | | |   (∀r:rel().
. . . | | |    r ∈ I
. . . | | |    ⇒
. . . | | |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | |                                         te);act))
. . . | | 26. r : rel()
. . . | | 27. r ∈ I
. . . | | 28. act ∈ (Σ[[A.da]] rho)
. . . | | 29. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | |                                        te);act)
. . . | | |   ⇒
. . . | | |   [[wp_rel(A;kind(act);r)]] rho A.ds ... de e x ... ...
. . . | | 30. r@0 : rel()
. . . | | 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | 32. rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | | ⊢  [[r@0]] rho A.ds dec_lookup(A.da;...) de e x ... ...
. . . | | | by Decide affects_trace_rel(te;kind(act);r) THENA Auto
. . . | | |
. . . | | |\
. . . | | | | 1. A : ioa{i:l}()
. . . | | | | 2. I : Collection(rel())
. . . | | | | 3. rho : Decl
. . . | | | | 4. de : sig()
. . . | | | | 5. e : {[[de]] rho}
. . . | | | | 6. te : Label→Label→𝔹
. . . | | | | 7. tc_ioa(A;de)
. . . | | | | 8. ¬ioa_mentions_trace(A)
. . . | | | | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | | | 10. tc_pred(I;A.ds;<>;de)
. . . | | | | 11. covers_pred(A;I)
. . . | | | | 12. guarded_trace(A.da;te;I)
. . . | | | | 13. closed_pred(I)
. . . | | | | 14. single_valued_decls(A.ds)
. . . | | | | 15. s0 : [[A]] rho de e.state
. . . | | | | 16. x : [[A]] rho de e.state
. . . | | | | 17. act : [[A]] rho de e.action
. . . | | | | 18. x' : [[A]] rho de e.state
. . . | | | | 19. tr : (Σ[[A.da]] rho) List
. . . | | | | 20. [[A]] rho de e.init(s0)
. . . | | | | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | | | | |                                       te).trace;x)
. . . | | | | 22. ∀r:rel().
. . . | | | | |   r ∈ I
```

```
. . . | | | |    ⇒
. . . | | | |    [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | | 23. [[A]] rho de e.trans(x,act,x')
. . . | | | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | | |    ⇒
. . . | | | |    (∀r:rel().
. . . | | | |     r ∈ I
. . . | | | |     ⇒
. . . | | | |     [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | | |                                            te);act))
. . . | | | 25. (∀r:rel().
. . . | | | |     (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | | |     ⇒
. . . | | | |     [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...)
. . . | | | |    ⇒
. . . | | | |    (∀r:rel().
. . . | | | |     r ∈ I
. . . | | | |     ⇒
. . . | | | |     [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | | |                                            te);act))
. . . | | | 26. r : rel()
. . . | | | 27. r ∈ I
. . . | | | 28. act ∈ (Σ[[A.da]] rho)
. . . | | | 29. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | | |                                          te);act)
. . . | | | |    ⇒
. . . | | | |    [[wp_rel(A;kind(act);r)]] rho A.ds ... de e x ... ...
. . . | | | 30. r@0 : rel()
. . . | | | 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | | 32. rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | | | 33. affects_trace_rel(te;kind(act);r)
. . . | | | ⊢  [[r@0]] rho A.ds dec_lookup(A.da;...) de e x ... ...
. . . | | | | by Unfold `wp_rel` (-3) THEN D (-10)
. . . | | | |
. . . | | | |\
. . . | | | | .....antecedent .....
. . . | | | | 1. A : ioa{i:l}()
. . . | | | | 2. I : Collection(rel())
. . . | | | | 3. rho : Decl
. . . | | | | 4. de : sig()
. . . | | | | 5. e : {[[de]] rho}
. . . | | | | 6. te : Label→Label→𝔹
. . . | | | | 7. tc_ioa(A;de)
. . . | | | | 8. ¬ioa_mentions_trace(A)
. . . | | | | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | | | 10. tc_pred(I;A.ds;<>;de)
. . . | | | | 11. covers_pred(A;I)
. . . | | | | 12. guarded_trace(A.da;te;I)
. . . | | | | 13. closed_pred(I)
. . . | | | | 14. single_valued_decls(A.ds)
. . . | | | | 15. s0 : [[A]] rho de e.state
. . . | | | | 16. x : [[A]] rho de e.state
. . . | | | | 17. act : [[A]] rho de e.action
. . . | | | | 18. x' : [[A]] rho de e.state
. . . | | | | 19. tr : (Σ[[A.da]] rho) List
```

```
. . . | | | | 20. [[A]] rho de e.init(s0)
. . . | | | | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | | | | |                                 te).trace;x)
. . . | | | | 22. ∀r:rel().
. . . | | | | |   r ∈ I
. . . | | | | |   ⇒
. . . | | | | |   [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | | | 23. [[A]] rho de e.trans(x,act,x')
. . . | | | | 24. (∀r:rel().
. . . | | | | |    (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | | | |    ⇒
. . . | | | | |    [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...)
. . . | | | | |   ⇒
. . . | | | | |   (∀r:rel().
. . . | | | | |    r ∈ I
. . . | | | | |    ⇒
. . . | | | | |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | | | 25. r : rel()
. . . | | | | 26. r ∈ I
. . . | | | | 27. act ∈ (Σ[[A.da]] rho)
. . . | | | | 28. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | | | |                                     te);act)
. . . | | | | |   ⇒
. . . | | | | |   [[wp_rel(A;kind(act);r)]] rho ... ... de e x ... ...
. . . | | | | 29. r@0 : rel()
. . . | | | | 30. r@0 ∈ smts_eff_rel(action_effect(...;A.eff;...);r)
. . . | | | | 31. rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | | | | 32. affects_trace_rel(te;kind(act);r)
. . . | | | | ⊢  ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
. . . | | | | | by Assert guarded_trace(A.da;te;I)
. . . | | | | |
. . . | | | | |\
. . . | | | | | .....assertion .....
. . . | | | | | 1. A : ioa{i:l}()
. . . | | | | | 2. I : Collection(rel())
. . . | | | | | 3. rho : Decl
. . . | | | | | 4. de : sig()
. . . | | | | | 5. e : {[[de]] rho}
. . . | | | | | 6. te : Label→Label→𝔹
. . . | | | | | 7. tc_ioa(A;de)
. . . | | | | | 8. ¬ioa_mentions_trace(A)
. . . | | | | | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | | | | 10. tc_pred(I;A.ds;<>;de)
. . . | | | | | 11. covers_pred(A;I)
. . . | | | | | 12. guarded_trace(A.da;te;I)
. . . | | | | | 13. closed_pred(I)
. . . | | | | | 14. single_valued_decls(A.ds)
. . . | | | | | 15. s0 : [[A]] rho de e.state
. . . | | | | | 16. x : [[A]] rho de e.state
. . . | | | | | 17. act : [[A]] rho de e.action
. . . | | | | | 18. x' : [[A]] rho de e.state
. . . | | | | | 19. tr : (Σ[[A.da]] rho) List
. . . | | | | | 20. [[A]] rho de e.init(s0)
. . . | | | | | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | | | | |                                     te).trace;x)
```

```
. . . | | | | | 22. ∀r:rel().
. . . | | | | |     r ∈ I
. . . | | | | |     ⇒
. . . | | | | |     [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | | | | 23. [[A]] rho de e.trans(x,act,x')
. . . | | | | | 24. ...
. . . | | | | |     ⇒
. . . | | | | |     (∀r:rel().
. . . | | | | |      r ∈ I
. . . | | | | |      ⇒
. . . | | | | |      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | | | | 25. r : rel()
. . . | | | | | 26. r ∈ I
. . . | | | | | 27. act ∈ (Σ[[A.da]] rho)
. . . | | | | | 28. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | | | |     ⇒
. . . | | | | |     [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | | | | 29. r@0 : rel()
. . . | | | | | 30. r@0 ∈ smts_eff_rel(action_effect(...;A.eff;...);r)
. . . | | | | | 31. rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | | | | | 32. affects_trace_rel(te;kind(act);r)
. . . | | | | | ⊢  guarded_trace(A.da;te;I)  by Trivial
. . . | | | | |
. . . | | | | | .\
. . . | | | | | . 1. A : ioa{i:l}()
. . . | | | | | . 2. I : Collection(rel())
. . . | | | | | . 3. rho : Decl
. . . | | | | | . 4. de : sig()
. . . | | | | | . 5. e : {[[de]] rho}
. . . | | | | | . 6. te : Label→Label→𝔹
. . . | | | | | . 7. tc_ioa(A;de)
. . . | | | | | . 8. ¬ioa_mentions_trace(A)
. . . | | | | | . 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | | | | . 10. tc_pred(I;A.ds;<>;de)
. . . | | | | | . 11. covers_pred(A;I)
. . . | | | | | . 12. guarded_trace(A.da;te;I)
. . . | | | | | . 13. closed_pred(I)
. . . | | | | | . 14. single_valued_decls(A.ds)
. . . | | | | | . 15. s0 : [[A]] rho de e.state
. . . | | | | | . 16. x : [[A]] rho de e.state
. . . | | | | | . 17. act : [[A]] rho de e.action
. . . | | | | | . 18. x' : [[A]] rho de e.state
. . . | | | | | . 19. tr : (Σ[[A.da]] rho) List
. . . | | | | | . 20. [[A]] rho de e.init(s0)
. . . | | | | | . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | | | | | .                                    te).trace;x)
. . . | | | | | . 22. ∀r:rel().
. . . | | | | | .     r ∈ I
. . . | | | | | .     ⇒
. . . | | | | | .     [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | | | | . 23. [[A]] rho de e.trans(x,act,x')
. . . | | | | | . 24. ...
. . . | | | | | .     ⇒
. . . | | | | | .     (∀r:rel().
. . . | | | | | .      r ∈ I
```

```
. . . | | | | .      ⇒
. . . | | | | .      [[r]] rho A.ds <> de e x’ · tappend(...;act))
. . . | | | | . 25. r : rel()
. . . | | | | . 26. r ∈ I
. . . | | | | . 27. act ∈ (Σ[[A.da]] rho)
. . . | | | | . 28. [[r]] rho A.ds <> de e x’ · tappend(...;act)
. . . | | | | .     ⇒
. . . | | | | .     [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | | | . 29. r@0 : rel()
. . . | | | | . 30. r@0 ∈ smts_eff_rel(action_effect(...;A.eff;...);r)
. . . | | | | . 31. rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | | | | . 32. affects_trace_rel(te;kind(act);r)
. . . | | | | . 33. guarded_trace(A.da;te;I)
. . . | | | | . ⊢  ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
. . . | | | | .   by Unfold ‘guarded_trace‘ (-1)
. . . | | | | . THEN InstHyp [r
. . . | | | | .      ] (-1) THENA Auto
. . . | | | | .
. . . | | | | .      THEN BackThruHyp (-1) THEN Auto
. . . | | | |
. . . | | | .\
. . . | | | . 1. A : ioa{i:l}()
. . . | | | . 2. I : Collection(rel())
. . . | | | . 3. rho : Decl
. . . | | | . 4. de : sig()
. . . | | | . 5. e : {[[de]] rho}
. . . | | | . 6. te : Label→Label→𝔹
. . . | | | . 7. tc_ioa(A;de)
. . . | | | . 8. ¬ioa_mentions_trace(A)
. . . | | | . 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | | . 10. tc_pred(I;A.ds;<>;de)
. . . | | | . 11. covers_pred(A;I)
. . . | | | . 12. guarded_trace(A.da;te;I)
. . . | | | . 13. closed_pred(I)
. . . | | | . 14. single_valued_decls(A.ds)
. . . | | | . 15. s0 : [[A]] rho de e.state
. . . | | | . 16. x : [[A]] rho de e.state
. . . | | | . 17. act : [[A]] rho de e.action
. . . | | | . 18. x’ : [[A]] rho de e.state
. . . | | | . 19. tr : (Σ[[A.da]] rho) List
. . . | | | . 20. [[A]] rho de e.init(s0)
. . . | | | . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | | | . |                                 te).trace;x)
. . . | | | . 22. ∀r:rel().
. . . | | | . |   r ∈ I
. . . | | | . |   ⇒
. . . | | | . |   [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | | . 23. [[A]] rho de e.trans(x,act,x’)
. . . | | | . 24. (∀r:rel().
. . . | | | . |    (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | | . |    ⇒
. . . | | | . |    [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...)
. . . | | | . |   ⇒
. . . | | | . |   (∀r:rel().
. . . | | | . |    r ∈ I
```

```
. . . | | | . |     ⇒
. . . | | | . |     [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | | . 25. r : rel()
. . . | | | . 26. r ∈ I
. . . | | | . 27. act ∈ (Σ[[A.da]] rho)
. . . | | | . 28. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | | . |                                         te);act)
. . . | | | . |   ⇒
. . . | | | . |   [[wp_rel(A;kind(act);r)]] rho ... ... de e x ... ...
. . . | | | . 29. r@0 : rel()
. . . | | | . 30. r@0 ∈ smts_eff_rel(action_effect(...;A.eff;...);r)
. . . | | | . 31. rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | | | . 32. affects_trace_rel(te;kind(act);r)
. . . | | | . 33. ∀r:rel().
. . . | | | . |   r ∈ I
. . . | | | . |   ⇒
. . . | | | . |   [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | | . |                                         te);act)
. . . | | | . ⊢  [[r@0]] rho A.ds dec_lookup(A.da;...) de e x ... ...
. . . | | | . | by InstHyp [r] (-1) THENA Auto
. . . | | | . |
. . . | | | . 1. A : ioa{i:l}()
. . . | | | . 2. I : Collection(rel())
. . . | | | . 3. rho : Decl
. . . | | | . 4. de : sig()
. . . | | | . 5. e : {[[de]] rho}
. . . | | | . 6. te : Label→Label→𝔹
. . . | | | . 7. tc_ioa(A;de)
. . . | | | . 8. ¬ioa_mentions_trace(A)
. . . | | | . 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | | . 10. tc_pred(I;A.ds;<>;de)
. . . | | | . 11. covers_pred(A;I)
. . . | | | . 12. guarded_trace(A.da;te;I)
. . . | | | . 13. closed_pred(I)
. . . | | | . 14. single_valued_decls(A.ds)
. . . | | | . 15. s0 : [[A]] rho de e.state
. . . | | | . 16. x : [[A]] rho de e.state
. . . | | | . 17. act : [[A]] rho de e.action
. . . | | | . 18. x' : [[A]] rho de e.state
. . . | | | . 19. tr : (Σ[[A.da]] rho) List
. . . | | | . 20. [[A]] rho de e.init(s0)
. . . | | | . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | | | . |                                     te).trace;x)
. . . | | | . 22. ∀r:rel().
. . . | | | . |   r ∈ I
. . . | | | . |   ⇒
. . . | | | . |   [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | | . 23. [[A]] rho de e.trans(x,act,x')
. . . | | | . 24. (∀r:rel().
. . . | | | . |     (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | | . |     ⇒
. . . | | | . |     [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...)
. . . | | | . |   ⇒
. . . | | | . |   (∀r:rel().
. . . | | | . |    r ∈ I
```

```
. . . | | | . |    ⇒
. . . | | | . |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | | . 25. r : rel()
. . . | | | . 26. r ∈ I
. . . | | | . 27. act ∈ (Σ[[A.da]] rho)
. . . | | | . 28. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | | . |                                             te);act)
. . . | | | . |   ⇒
. . . | | | . |   [[wp_rel(A;kind(act);r)]] rho ... ... de e x ... ...
. . . | | | . 29. r@0 : rel()
. . . | | | . 30. r@0 ∈ smts_eff_rel(action_effect(...;A.eff;...);r)
. . . | | | . 31. rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | | | . 32. affects_trace_rel(te;kind(act);r)
. . . | | | . 33. ∀r:rel().
. . . | | | . |   r ∈ I
. . . | | | . |   ⇒
. . . | | | . |   [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | | . |                                             te);act)
. . . | | | . 34. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | | . |                                             te);act)
. . . | | | . ⊢  [[r@0]] rho A.ds dec_lookup(A.da;...) de e x ... ...
. . . | | | . | by Assert
. . . | | | . |    [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | | . |
. . . | | | . |
. . . | | | . |\
. . . | | | . | .....assertion .....
. . . | | | . | 1. A : ioa{i:l}()
. . . | | | . | 2. I : Collection(rel())
. . . | | | . | 3. rho : Decl
. . . | | | . | 4. de : sig()
. . . | | | . | 5. e : {[[de]] rho}
. . . | | | . | 6. te : Label→Label→𝔹
. . . | | | . | 7. tc_ioa(A;de)
. . . | | | . | 8. ¬ioa_mentions_trace(A)
. . . | | | . | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | | . | 10. tc_pred(I;A.ds;<>;de)
. . . | | | . | 11. covers_pred(A;I)
. . . | | | . | 12. guarded_trace(A.da;te;I)
. . . | | | . | 13. closed_pred(I)
. . . | | | . | 14. single_valued_decls(A.ds)
. . . | | | . | 15. s0 : [[A]] rho de e.state
. . . | | | . | 16. x : [[A]] rho de e.state
. . . | | | . | 17. act : [[A]] rho de e.action
. . . | | | . | 18. x' : [[A]] rho de e.state
. . . | | | . | 19. tr : (Σ[[A.da]] rho) List
. . . | | | . | 20. [[A]] rho de e.init(s0)
. . . | | | . | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | | | . | |                                           te).trace;x)
. . . | | | . | 22. ∀r:rel().
. . . | | | . | |   r ∈ I
. . . | | | . | |   ⇒
. . . | | | . | |   [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | | . | 23. [[A]] rho de e.trans(x,act,x')
. . . | | | . | 24. ...
```

```
. . . | | | . | |   ⇒
. . . | | | . | |   (∀r:rel().
. . . | | | . | |    r ∈ I
. . . | | | . | |    ⇒
. . . | | | . | |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | | . | 25. r : rel()
. . . | | | . | 26. r ∈ I
. . . | | | . | 27. act ∈ (Σ[[A.da]] rho)
. . . | | | . | 28. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | | . | |   ⇒
. . . | | | . | |   [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | | . | 29. r@0 : rel()
. . . | | | . | 30. r@0 ∈ smts_eff_rel(action_effect(...;A.eff;...);r)
. . . | | | . | 31. rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | | | . | 32. affects_trace_rel(te;kind(act);r)
. . . | | | . | 33. ∀r:rel().
. . . | | | . | |   r ∈ I
. . . | | | . | |   ⇒
. . . | | | . | |   [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | | . | 34. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | | . | ⊢  [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | | . | | by  InstLemma `wp_rel_correctness` [A;r;rho;de;e;act
. . . | | | . | |    ;tappend(mk_trace_env(tr, te);act);x;x'
. . . | | | . | |    ] THENA (Auto THEN TrivialIoaHyp)
. . . | | | . |
. . . | | | . | |    THEN EasyHypThen Auto
. . . | | | . |
. . . | | | . | |    THEN Try (ThinTrivial THEN Trivial)
. . . | | | . |
. . . | | | . | |    ...
. . . | | | . | |
. . . | | | . | 1. A : ioa{i:l}()
. . . | | | . | 2. I : Collection(rel())
. . . | | | . | 3. rho : Decl
. . . | | | . | 4. de : sig()
. . . | | | . | 5. e : {[[de]] rho}
. . . | | | . | 6. te : Label→Label→𝔹
. . . | | | . | 7. tc_ioa(A;de)
. . . | | | . | 8. ¬ioa_mentions_trace(A)
. . . | | | . | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | | . | 10. tc_pred(I;A.ds;<>;de)
. . . | | | . | 11. covers_pred(A;I)
. . . | | | . | 12. guarded_trace(A.da;te;I)
. . . | | | . | 13. closed_pred(I)
. . . | | | . | 14. single_valued_decls(A.ds)
. . . | | | . | 15. s0 : [[A]] rho de e.state
. . . | | | . | 16. x : [[A]] rho de e.state
. . . | | | . | 17. act : [[A]] rho de e.action
. . . | | | . | 18. x' : [[A]] rho de e.state
. . . | | | . | 19. tr : (Σ[[A.da]] rho) List
. . . | | | . | 20. [[A]] rho de e.init(s0)
. . . | | | . | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | | | . |                                       te).trace;x)
. . . | | | . | 22. ∀r:rel().
. . . | | | . |     r ∈ I
```

```
. . . | | | . |     ⇒
. . . | | | . |     [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | | . | 23. [[A]] rho de e.trans(x,act,x')
. . . | | | . | 24. ...
. . . | | | . |     ⇒
. . . | | | . |     (∀r:rel().
. . . | | | . |      r ∈ I
. . . | | | . |      ⇒
. . . | | | . |      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | | . | 25. r : rel()
. . . | | | . | 26. r ∈ I
. . . | | | . | 27. act ∈ (Σ[[A.da]] rho)
. . . | | | . | 28. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | | . |     ⇒
. . . | | | . |     [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | | . | 29. r@0 : rel()
. . . | | | . | 30. r@0 ∈ smts_eff_rel(action_effect(...;A.eff;...);r)
. . . | | | . | 31. rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | | | . | 32. affects_trace_rel(te;kind(act);r)
. . . | | | . | 33. ∀r:rel().
. . . | | | . |     r ∈ I
. . . | | | . |     ⇒
. . . | | | . |     [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | | . | 34. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | | . | ⊢  covers_rel(A;r)  by Easy
. . . | | | . |
. . . | | | . .\
. . . | | | . . 1. A : ioa{i:l}()
. . . | | | . . 2. I : Collection(rel())
. . . | | | . . 3. rho : Decl
. . . | | | . . 4. de : sig()
. . . | | | . . 5. e : {[[de]] rho}
. . . | | | . . 6. te : Label→Label→𝔹
. . . | | | . . 7. tc_ioa(A;de)
. . . | | | . . 8. ¬ioa_mentions_trace(A)
. . . | | | . . 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | | . . 10. tc_pred(I;A.ds;<>;de)
. . . | | | . . 11. covers_pred(A;I)
. . . | | | . . 12. guarded_trace(A.da;te;I)
. . . | | | . . 13. closed_pred(I)
. . . | | | . . 14. single_valued_decls(A.ds)
. . . | | | . . 15. s0 : [[A]] rho de e.state
. . . | | | . . 16. x : [[A]] rho de e.state
. . . | | | . . 17. act : [[A]] rho de e.action
. . . | | | . . 18. x' : [[A]] rho de e.state
. . . | | | . . 19. tr : (Σ[[A.da]] rho) List
. . . | | | . . 20. [[A]] rho de e.init(s0)
. . . | | | . . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | | | . .                                         te).trace;x)
. . . | | | . . 22. ∀r:rel().
. . . | | | . .     r ∈ I
. . . | | | . .     ⇒
. . . | | | . .     [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | | . . 23. [[A]] rho de e.trans(x,act,x')
. . . | | | . . 24. ...
```

```
. . . | | | . .     ⇒
. . . | | | . .     (∀r:rel().
. . . | | | . .      r ∈ I
. . . | | | . .      ⇒
. . . | | | . .      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | | . . 25. r : rel()
. . . | | | . . 26. r ∈ I
. . . | | | . . 27. act ∈ (Σ[[A.da]] rho)
. . . | | | . . 28. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | | . .     ⇒
. . . | | | . .     [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | | . . 29. r@0 : rel()
. . . | | | . . 30. r@0 ∈ smts_eff_rel(action_effect(...;A.eff;...);r)
. . . | | | . . 31. rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | | | . . 32. affects_trace_rel(te;kind(act);r)
. . . | | | . . 33. ∀r:rel().
. . . | | | . .     r ∈ I
. . . | | | . .     ⇒
. . . | | | . .     [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | | . . 34. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | | . . 35. [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | | . . ⊢  [[r@0]] rho A.ds dec_lookup(...;...) de e x ... ...
. . . | | | . .   by ...
. . . | | |
. . . | | .\
. . . | | . 1. A : ioa{i:l}()
. . . | | . 2. I : Collection(rel())
. . . | | . 3. rho : Decl
. . . | | . 4. de : sig()
. . . | | . 5. e : {[[de]] rho}
. . . | | . 6. te : Label→Label→𝔹
. . . | | . 7. tc_ioa(A;de)
. . . | | . 8. ¬ioa_mentions_trace(A)
. . . | | . 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | . 10. tc_pred(I;A.ds;<>;de)
. . . | | . 11. covers_pred(A;I)
. . . | | . 12. guarded_trace(A.da;te;I)
. . . | | . 13. closed_pred(I)
. . . | | . 14. single_valued_decls(A.ds)
. . . | | . 15. s0 : [[A]] rho de e.state
. . . | | . 16. x : [[A]] rho de e.state
. . . | | . 17. act : [[A]] rho de e.action
. . . | | . 18. x' : [[A]] rho de e.state
. . . | | . 19. tr : (Σ[[A.da]] rho) List
. . . | | . 20. [[A]] rho de e.init(s0)
. . . | | . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | | . |                                    te).trace;x)
. . . | | . 22. ∀r:rel().
. . . | | . |   r ∈ I
. . . | | . |   ⇒
. . . | | . |   [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | . 23. [[A]] rho de e.trans(x,act,x')
. . . | | . 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | . |   ⇒
. . . | | . |   (∀r:rel().
```

```
. . . | | . |    r ∈ I
. . . | | . |    ⇒
. . . | | . |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | . |                                              te);act))
. . . | | . 25. (∀r:rel().
. . . | | . |    (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | . |    ⇒
. . . | | . |    [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...)
. . . | | . |   ⇒
. . . | | . |   (∀r:rel().
. . . | | . |    r ∈ I
. . . | | . |    ⇒
. . . | | . |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | . |                                              te);act))
. . . | | . 26. r : rel()
. . . | | . 27. r ∈ I
. . . | | . 28. act ∈ (Σ[[A.da]] rho)
. . . | | . 29. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | . |                                             te);act)
. . . | | . |   ⇒
. . . | | . |   [[wp_rel(A;kind(act);r)]] rho A.ds ... de e x ... ...
. . . | | . 30. r@0 : rel()
. . . | | . 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | . 32. rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | | . 33. ¬affects_trace_rel(te;kind(act);r)
. . . | | . ⊢  [[r@0]] rho A.ds dec_lookup(A.da;...) de e x ... ...
. . . | | . | by Assert
. . . | | . |    [[rel_unprime(r@0)]] rho A.ds ... de e x ... ...
. . . | | . |
. . . | | . |\
. . . | | . | .....assertion .....
. . . | | . | 1. A : ioa{i:l}()
. . . | | . | 2. I : Collection(rel())
. . . | | . | 3. rho : Decl
. . . | | . | 4. de : sig()
. . . | | . | 5. e : {[[de]] rho}
. . . | | . | 6. te : Label→Label→𝔹
. . . | | . | 7. tc_ioa(A;de)
. . . | | . | 8. ¬ioa_mentions_trace(A)
. . . | | . | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | . | 10. tc_pred(I;A.ds;<>;de)
. . . | | . | 11. covers_pred(A;I)
. . . | | . | 12. guarded_trace(A.da;te;I)
. . . | | . | 13. closed_pred(I)
. . . | | . | 14. single_valued_decls(A.ds)
. . . | | . | 15. s0 : [[A]] rho de e.state
. . . | | . | 16. x : [[A]] rho de e.state
. . . | | . | 17. act : [[A]] rho de e.action
. . . | | . | 18. x' : [[A]] rho de e.state
. . . | | . | 19. tr : (Σ[[A.da]] rho) List
. . . | | . | 20. [[A]] rho de e.init(s0)
. . . | | . | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | | . | |                                           te).trace;x)
. . . | | . | 22. ∀r:rel().
. . . | | . | |   r ∈ I
```

```
. . . | | . | |   ⇒
. . . | | . | |   [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | . | 23. [[A]] rho de e.trans(x,act,x')
. . . | | . | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | . | |   ⇒
. . . | | . | |   (∀r:rel().
. . . | | . | |    r ∈ I
. . . | | . | |    ⇒
. . . | | . | |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | 25. (∀r:rel().
. . . | | . | |    (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | . | |    ⇒
. . . | | . | |    [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...)
. . . | | . | |   ⇒
. . . | | . | |   (∀r:rel().
. . . | | . | |    r ∈ I
. . . | | . | |    ⇒
. . . | | . | |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | 26. r : rel()
. . . | | . | 27. r ∈ I
. . . | | . | 28. act ∈ (Σ[[A.da]] rho)
. . . | | . | 29. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | . | |                                          te);act)
. . . | | . | |   ⇒
. . . | | . | |   [[wp_rel(A;kind(act);r)]] rho ... ... de e x ... ...
. . . | | . | 30. r@0 : rel()
. . . | | . | 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | . | 32. rel_eq(rel_unprime(r@0);rel_unprime(r))

. . . | | . | 33. ¬affects_trace_rel(te;kind(act);r)
. . . | | . | ⊢  [[rel_unprime(r@0)]] rho A.ds ... de e x ... ...
. . . | | . | | by ...
. . . | | . | |
. . . | | . | 1. A : ioa{i:l}()
. . . | | . | 2. I : Collection(rel())
. . . | | . | 3. rho : Decl
. . . | | . | 4. de : sig()
. . . | | . | 5. e : {[[de]] rho}
. . . | | . | 6. te : Label→Label→𝔹
. . . | | . | 7. tc_ioa(A;de)
. . . | | . | 8. ¬ioa_mentions_trace(A)
. . . | | . | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | . | 10. tc_pred(I;A.ds;<>;de)
. . . | | . | 11. covers_pred(A;I)
. . . | | . | 12. guarded_trace(A.da;te;I)
. . . | | . | 13. closed_pred(I)
. . . | | . | 14. single_valued_decls(A.ds)
. . . | | . | 15. s0 : [[A]] rho de e.state
. . . | | . | 16. x : [[A]] rho de e.state
. . . | | . | 17. act : [[A]] rho de e.action
. . . | | . | 18. x' : [[A]] rho de e.state
. . . | | . | 19. tr : (Σ[[A.da]] rho) List
. . . | | . | 20. [[A]] rho de e.init(s0)
. . . | | . | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | | . | |                                        te).trace;x)
. . . | | . | 22. ∀r:rel().
```

```
. . . | | . | |   r ∈ I
. . . | | . | |   ⇒
. . . | | . | |   [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | . | 23. [[A]] rho de e.trans(x,act,x')
. . . | | . | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | . | |   ⇒
. . . | | . | |   (∀r:rel().
. . . | | . | |    r ∈ I
. . . | | . | |    ⇒
. . . | | . | |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | 25. (∀r:rel().
. . . | | . | |    (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | . | |    ⇒
. . . | | . | |    [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...)
. . . | | . | |   ⇒
. . . | | . | |   (∀r:rel().
. . . | | . | |    r ∈ I
. . . | | . | |    ⇒
. . . | | . | |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | 26. r : rel()
. . . | | . | 27. r ∈ I
. . . | | . | 28. act ∈ (Σ[[A.da]] rho)
. . . | | . | 29. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | . | |                                          te);act)
. . . | | . | |   ⇒
. . . | | . | |   [[wp_rel(A;kind(act);r)]] rho ... ... de e x ... ...
. . . | | . | 30. r@0 : rel()
. . . | | . | 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | . | 32. rel_unprime(r@0) = rel_unprime(r)
. . . | | . | 33. ¬affects_trace_rel(te;kind(act);r)
. . . | | . | ⊢  [[rel_unprime(r)]] rho A.ds ... de e x value(act) ...
. . . | | . | | by Assert
. . . | | . | |    [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...
. . . | | . | |
. . . | | . | |\
. . . | | . | | .....assertion .....
. . . | | . | | 1. A : ioa{i:l}()
. . . | | . | | 2. I : Collection(rel())
. . . | | . | | 3. rho : Decl
. . . | | . | | 4. de : sig()
. . . | | . | | 5. e : {[[de]] rho}
. . . | | . | | 6. te : Label→Label→𝔹
. . . | | . | | 7. tc_ioa(A;de)
. . . | | . | | 8. ¬ioa_mentions_trace(A)
. . . | | . | | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | . | | 10. tc_pred(I;A.ds;<>;de)
. . . | | . | | 11. covers_pred(A;I)
. . . | | . | | 12. guarded_trace(A.da;te;I)
. . . | | . | | 13. closed_pred(I)
. . . | | . | | 14. single_valued_decls(A.ds)
. . . | | . | | 15. s0 : [[A]] rho de e.state
. . . | | . | | 16. x : [[A]] rho de e.state
. . . | | . | | 17. act : [[A]] rho de e.action
. . . | | . | | 18. x' : [[A]] rho de e.state
. . . | | . | | 19. tr : (Σ[[A.da]] rho) List
```

```
. . . | | . | | 20. [[A]] rho de e.init(s0)
. . . | | . | | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | | . | | |                                  te).trace;x)
. . . | | . | | 22. ∀r:rel().
. . . | | . | | |   r ∈ I
. . . | | . | | |   ⇒
. . . | | . | | |   [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | . | | 23. [[A]] rho de e.trans(x,act,x')
. . . | | . | | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | . | | |   ⇒
. . . | | . | | |   (∀r:rel().
. . . | | . | | |    r ∈ I
. . . | | . | | |    ⇒
. . . | | . | | |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | | 25. ...
. . . | | . | | |   ⇒
. . . | | . | | |   (∀r:rel().
. . . | | . | | |    r ∈ I
. . . | | . | | |    ⇒
. . . | | . | | |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | | 26. r : rel()
. . . | | . | | 27. r ∈ I
. . . | | . | | 28. act ∈ (Σ[[A.da]] rho)
. . . | | . | | 29. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | . | | |   ⇒
. . . | | . | | |   [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | . | | 30. r@0 : rel()
. . . | | . | | 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | . | | 32. rel_unprime(r@0) = rel_unprime(r)
. . . | | . | | 33. ¬affects_trace_rel(te;kind(act);r)
. . . | | . | | ⊢  [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...
. . . | | . | | | by AllHyps (\i. InstHyp [r] i THENA (Complete Auto))
. . . | | . | | |
. . . | | . | | |
. . . | | . | | 1. A : ioa{i:l}()
. . . | | . | | 2. I : Collection(rel())
. . . | | . | | 3. rho : Decl
. . . | | . | | 4. de : sig()
. . . | | . | | 5. e : {[[de]] rho}
. . . | | . | | 6. te : Label→Label→𝔹
. . . | | . | | 7. tc_ioa(A;de)
. . . | | . | | 8. ¬ioa_mentions_trace(A)
. . . | | . | | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | . | | 10. tc_pred(I;A.ds;<>;de)
. . . | | . | | 11. covers_pred(A;I)
. . . | | . | | 12. guarded_trace(A.da;te;I)
. . . | | . | | 13. closed_pred(I)
. . . | | . | | 14. single_valued_decls(A.ds)
. . . | | . | | 15. s0 : [[A]] rho de e.state
. . . | | . | | 16. x : [[A]] rho de e.state
. . . | | . | | 17. act : [[A]] rho de e.action
. . . | | . | | 18. x' : [[A]] rho de e.state
. . . | | . | | 19. tr : (Σ[[A.da]] rho) List
. . . | | . | | 20. [[A]] rho de e.init(s0)
. . . | | . | | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
```

```
. . . | | . | | |                                     te).trace;x)
. . . | | . | | 22. ∀r:rel().
. . . | | . | | |   r ∈ I
. . . | | . | | |   ⇒
. . . | | . | | |   [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | . | | 23. [[A]] rho de e.trans(x,act,x')
. . . | | . | | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | . | | |   ⇒
. . . | | . | | |   (∀r:rel().
. . . | | . | | |    r ∈ I
. . . | | . | | |    ⇒
. . . | | . | | |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | | 25. ...
. . . | | . | | |   ⇒
. . . | | . | | |   (∀r:rel().
. . . | | . | | |    r ∈ I
. . . | | . | | |    ⇒
. . . | | . | | |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | | 26. r : rel()
. . . | | . | | 27. r ∈ I
. . . | | . | | 28. act ∈ (Σ[[A.da]] rho)
. . . | | . | | 29. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | . | | |   ⇒
. . . | | . | | |   [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | . | | 30. r@0 : rel()
. . . | | . | | 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | . | | 32. rel_unprime(r@0) = rel_unprime(r)
. . . | | . | | 33. ¬affects_trace_rel(te;kind(act);r)
. . . | | . | | 34. [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | . | | ⊢  [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...
. . . | | . | | | by BackThruLemma' `closed_rel_mng_2` THEN Auto
. . . | | . | |
. . . | | . | | |    THEN Try (Unfold `decl` 0 THEN Auto)
. . . | | . | | |
. . . | | . | | |\
. . . | | . | | | .....wf .....
. . . | | . | | | 1. A : ioa{i:l}()
. . . | | . | | | 2. I : Collection(rel())
. . . | | . | | | 3. rho : Decl
. . . | | . | | | 4. de : sig()
. . . | | . | | | 5. e : {[[de]] rho}
. . . | | . | | | 6. te : Label→Label→𝔹
. . . | | . | | | 7. tc_ioa(A;de)
. . . | | . | | | 8. ¬ioa_mentions_trace(A)
. . . | | . | | | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | . | | | 10. tc_pred(I;A.ds;<>;de)
. . . | | . | | | 11. covers_pred(A;I)
. . . | | . | | | 12. guarded_trace(A.da;te;I)
. . . | | . | | | 13. closed_pred(I)
. . . | | . | | | 14. single_valued_decls(A.ds)
. . . | | . | | | 15. s0 : [[A]] rho de e.state
. . . | | . | | | 16. x : [[A]] rho de e.state
. . . | | . | | | 17. act : [[A]] rho de e.action
. . . | | . | | | 18. x' : [[A]] rho de e.state
. . . | | . | | | 19. tr : (Σ[[A.da]] rho) List
```

```
. . . | | . | | | 20. [[A]] rho de e.init(s0)
. . . | | . | | | 21. trace_reachable([[A]] rho de e;s0;....trace;x)
. . . | | . | | | 22. ∀r:rel().
. . . | | . | | |     r ∈ I
. . . | | . | | |     ⇒
. . . | | . | | |     [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | . | | | 23. [[A]] rho de e.trans(x,act,x')
. . . | | . | | | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | . | | |     ⇒
. . . | | . | | |     (∀r:rel().
. . . | | . | | |      r ∈ I
. . . | | . | | |      ⇒
. . . | | . | | |      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | | | 25. (∀r:rel().
. . . | | . | | |      (∃r@0:rel().
. . . | | . | | |       r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | . | | |      ⇒
. . . | | . | | |      [[r]] rho A.ds ... de e x value(act) ...)
. . . | | . | | |     ⇒
. . . | | . | | |     (∀r:rel().
. . . | | . | | |      r ∈ I
. . . | | . | | |      ⇒
. . . | | . | | |      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | | | 26. r : rel()
. . . | | . | | | 27. r ∈ I
. . . | | . | | | 28. act ∈ (Σ[[A.da]] rho)
. . . | | . | | | 29. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | . | | |     ⇒
. . . | | . | | |     [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | . | | | 30. r@0 : rel()
. . . | | . | | | 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | . | | | 32. rel_unprime(r@0) = rel_unprime(r)
. . . | | . | | | 33. ¬affects_trace_rel(te;kind(act);r)
. . . | | . | | | ⊢  x ∈ {[[A.ds]] rho}  by TrivialIoaHyp
. . . | | . | | |
. . . | | . | | |\
. . . | | . | | | 1. A : ioa{i:l}()
. . . | | . | | | 2. I : Collection(rel())
. . . | | . | | | 3. rho : Decl
. . . | | . | | | 4. de : sig()
. . . | | . | | | 5. e : {[[de]] rho}
. . . | | . | | | 6. te : Label→Label→𝔹
. . . | | . | | | 7. tc_ioa(A;de)
. . . | | . | | | 8. ¬ioa_mentions_trace(A)
. . . | | . | | | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | . | | | 10. tc_pred(I;A.ds;<>;de)
. . . | | . | | | 11. covers_pred(A;I)
. . . | | . | | | 12. guarded_trace(A.da;te;I)
. . . | | . | | | 13. closed_pred(I)
. . . | | . | | | 14. single_valued_decls(A.ds)
. . . | | . | | | 15. s0 : [[A]] rho de e.state
. . . | | . | | | 16. x : [[A]] rho de e.state
. . . | | . | | | 17. act : [[A]] rho de e.action
. . . | | . | | | 18. x' : [[A]] rho de e.state
. . . | | . | | | 19. tr : (Σ[[A.da]] rho) List
```

```
. . . | | . | | | 20. [[A]] rho de e.init(s0)
. . . | | . | | | 21. trace_reachable([[A]] rho de e;s0;....trace;x)
. . . | | . | | | 22. ∀r:rel().
. . . | | . | | | |   r ∈ I
. . . | | . | | | |   ⇒
. . . | | . | | | |   [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | . | | | 23. [[A]] rho de e.trans(x,act,x')
. . . | | . | | | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | . | | | |   ⇒
. . . | | . | | | |   (∀r:rel().
. . . | | . | | | |    r ∈ I
. . . | | . | | | |    ⇒
. . . | | . | | | |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | | | 25. (∀r:rel().
. . . | | . | | | |    (∃r@0:rel().
. . . | | . | | | |     r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | . | | | |    ⇒
. . . | | . | | | |    [[r]] rho A.ds ... de e x value(act) ...)
. . . | | . | | | |   ⇒
. . . | | . | | | |   (∀r:rel().
. . . | | . | | | |    r ∈ I
. . . | | . | | | |    ⇒
. . . | | . | | | |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | | | 26. r : rel()
. . . | | . | | | 27. r ∈ I
. . . | | . | | | 28. act ∈ (Σ[[A.da]] rho)
. . . | | . | | | 29. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | . | | | |   ⇒
. . . | | . | | | |   [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | . | | | 30. r@0 : rel()
. . . | | . | | | 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | . | | | 32. rel_unprime(r@0) = rel_unprime(r)
. . . | | . | | | 33. ¬affects_trace_rel(te;kind(act);r)
. . . | | . | | | ⊢  trace_consistent_rel(rho;A.da;mk_trace_env(tr,
. . . | | . | | | |                                   te).proj;r)
. . . | | . | | | | by Reduce 0 THEN Auto
. . . | | . | | | |
. . . | | . | | | 1. A : ioa{i:l}()
. . . | | . | | | 2. I : Collection(rel())
. . . | | . | | | 3. rho : Decl
. . . | | . | | | 4. de : sig()
. . . | | . | | | 5. e : {[[de]] rho}
. . . | | . | | | 6. te : Label→Label→𝔹
. . . | | . | | | 7. tc_ioa(A;de)
. . . | | . | | | 8. ¬ioa_mentions_trace(A)
. . . | | . | | | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | . | | | 10. tc_pred(I;A.ds;<>;de)
. . . | | . | | | 11. covers_pred(A;I)
. . . | | . | | | 12. guarded_trace(A.da;te;I)
. . . | | . | | | 13. closed_pred(I)
. . . | | . | | | 14. single_valued_decls(A.ds)
. . . | | . | | | 15. s0 : [[A]] rho de e.state
. . . | | . | | | 16. x : [[A]] rho de e.state

. . . | | . | | | 17. act : [[A]] rho de e.action
. . . | | . | | | 18. x' : [[A]] rho de e.state
```

```
. . . | | . | | | 19. tr : (Σ[[A.da]] rho) List
. . . | | . | | | 20. [[A]] rho de e.init(s0)
. . . | | . | | | 21. trace_reachable([[A]] rho de e;s0;....trace;x)
. . . | | . | | | 22. ∀r:rel().
. . . | | . | | |     r ∈ I
. . . | | . | | |     ⇒
. . . | | . | | |     [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | . | | | 23. [[A]] rho de e.trans(x,act,x')
. . . | | . | | | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | . | | |     ⇒
. . . | | . | | |     (∀r:rel().
. . . | | . | | |      r ∈ I
. . . | | . | | |      ⇒
. . . | | . | | |      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | | | 25. (∀r:rel().
. . . | | . | | |      (∃r@0:rel().
. . . | | . | | |       r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | . | | |      ⇒
. . . | | . | | |      [[r]] rho A.ds ... de e x value(act) ...)
. . . | | . | | |     ⇒
. . . | | . | | |     (∀r:rel().
. . . | | . | | |      r ∈ I
. . . | | . | | |      ⇒
. . . | | . | | |      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | | | 26. r : rel()
. . . | | . | | | 27. r ∈ I
. . . | | . | | | 28. act ∈ (Σ[[A.da]] rho)
. . . | | . | | | 29. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | . | | |     ⇒
. . . | | . | | |     [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | . | | | 30. r@0 : rel()
. . . | | . | | | 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | . | | | 32. rel_unprime(r@0) = rel_unprime(r)
. . . | | . | | | 33. ¬affects_trace_rel(te;kind(act);r)
. . . | | . | | | ⊢  trace_consistent_rel(rho;A.da;te;r)
. . . | | . | | |   by EasyHypThen Auto
. . . | | . | | |
. . . | | . | | |\
. . . | | . | | | 1. A : ioa{i:l}()
. . . | | . | | | 2. I : Collection(rel())
. . . | | . | | | 3. rho : Decl
. . . | | . | | | 4. de : sig()
. . . | | . | | | 5. e : {[[de]] rho}
. . . | | . | | | 6. te : Label→Label→𝔹
. . . | | . | | | 7. tc_ioa(A;de)
. . . | | . | | | 8. ¬ioa_mentions_trace(A)
. . . | | . | | | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | . | | | 10. tc_pred(I;A.ds;<>;de)
. . . | | . | | | 11. covers_pred(A;I)
. . . | | . | | | 12. guarded_trace(A.da;te;I)
. . . | | . | | | 13. closed_pred(I)
. . . | | . | | | 14. single_valued_decls(A.ds)
. . . | | . | | | 15. s0 : [[A]] rho de e.state
. . . | | . | | | 16. x : [[A]] rho de e.state
. . . | | . | | | 17. act : [[A]] rho de e.action
```

```
. . . | | . | | | 18. x' : [[A]] rho de e.state
. . . | | . | | | 19. tr : (Σ[[A.da]] rho) List
. . . | | . | | | 20. [[A]] rho de e.init(s0)
. . . | | . | | | 21. trace_reachable([[A]] rho de e;s0;....trace;x)
. . . | | . | | | 22. ∀r:rel().
. . . | | . | | |     r ∈ I
. . . | | . | | |     ⇒
. . . | | . | | |     [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | . | | | 23. [[A]] rho de e.trans(x,act,x')
. . . | | . | | | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | . | | |     ⇒
. . . | | . | | |     (∀r:rel().
. . . | | . | | |      r ∈ I
. . . | | . | | |      ⇒
. . . | | . | | |      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | | | 25. (∀r:rel().
. . . | | . | | |      (∃r@0:rel().
. . . | | . | | |       r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | . | | |      ⇒
. . . | | . | | |      [[r]] rho A.ds ... de e x value(act) ...)
. . . | | . | | |     ⇒
. . . | | . | | |     (∀r:rel().
. . . | | . | | |      r ∈ I
. . . | | . | | |      ⇒
. . . | | . | | |      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | | | 26. r : rel()
. . . | | . | | | 27. r ∈ I
. . . | | . | | | 28. act ∈ (Σ[[A.da]] rho)
. . . | | . | | | 29. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | . | | |     ⇒
. . . | | . | | |     [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | . | | | 30. r@0 : rel()
. . . | | . | | | 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | . | | | 32. rel_unprime(r@0) = rel_unprime(r)
. . . | | . | | | 33. ¬affects_trace_rel(te;kind(act);r)
. . . | | . | | | ⊢  tc(r;A.ds;dec_lookup(A.da;kind(act));de)
. . . | | . | | |   by Using ['da1',<>
. . . | | . | | |      ] (BackThruLemma 'tc_closed_rel') THEN Auto
. . . | | . | | |
. . . | | . | | |        THEN EasyHypThen Auto
. . . | | . | | |
. . . | | . | | .\
. . . | | . | | . 1. A : ioa{i:l}()
. . . | | . | | . 2. I : Collection(rel())
. . . | | . | | . 3. rho : Decl
. . . | | . | | . 4. de : sig()
. . . | | . | | . 5. e : {[[de]] rho}
. . . | | . | | . 6. te : Label→Label→𝔹
. . . | | . | | . 7. tc_ioa(A;de)
. . . | | . | | . 8. ¬ioa_mentions_trace(A)
. . . | | . | | . 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | . | | . 10. tc_pred(I;A.ds;<>;de)
. . . | | . | | . 11. covers_pred(A;I)
. . . | | . | | . 12. guarded_trace(A.da;te;I)
. . . | | . | | . 13. closed_pred(I)
```

```
. . . | | . | | . 14. single_valued_decls(A.ds)
. . . | | . | | . 15. s0 : [[A]] rho de e.state
. . . | | . | | . 16. x : [[A]] rho de e.state
. . . | | . | | . 17. act : [[A]] rho de e.action
. . . | | . | | . 18. x' : [[A]] rho de e.state
. . . | | . | | . 19. tr : (Σ[[A.da]] rho) List
. . . | | . | | . 20. [[A]] rho de e.init(s0)
. . . | | . | | . 21. trace_reachable([[A]] rho de e;s0;....trace;x)
. . . | | . | | . 22. ∀r:rel().
. . . | | . | | .     r ∈ I
. . . | | . | | .     ⇒
. . . | | . | | .     [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | . | | . 23. [[A]] rho de e.trans(x,act,x')
. . . | | . | | . 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | . | | .     ⇒
. . . | | . | | .     (∀r:rel().
. . . | | . | | .      r ∈ I
. . . | | . | | .      ⇒
. . . | | . | | .      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | | . 25. (∀r:rel().
. . . | | . | | .      (∃r@0:rel().
. . . | | . | | .       r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | . | | .      ⇒
. . . | | . | | .      [[r]] rho A.ds ... de e x value(act) ...)
. . . | | . | | .     ⇒
. . . | | . | | .     (∀r:rel().
. . . | | . | | .      r ∈ I
. . . | | . | | .      ⇒
. . . | | . | | .      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | | . 26. r : rel()
. . . | | . | | . 27. r ∈ I
. . . | | . | | . 28. act ∈ (Σ[[A.da]] rho)
. . . | | . | | . 29. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | . | | .     ⇒
. . . | | . | | .     [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | . | | . 30. r@0 : rel()
. . . | | . | | . 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | . | | . 32. rel_unprime(r@0) = rel_unprime(r)
. . . | | . | | . 33. ¬affects_trace_rel(te;kind(act);r)
. . . | | . | | . ⊢  closed_rel(r)  by EasyHypThen Auto
. . . | | . | |
. . . | | . | .\
. . . | | . | . 1. A : ioa{i:l}()
. . . | | . | . 2. I : Collection(rel())
. . . | | . | . 3. rho : Decl
. . . | | . | . 4. de : sig()
. . . | | . | . 5. e : {[[de]] rho}
. . . | | . | . 6. te : Label→Label→𝔹
. . . | | . | . 7. tc_ioa(A;de)
. . . | | . | . 8. ¬ioa_mentions_trace(A)
. . . | | . | . 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | . | . 10. tc_pred(I;A.ds;<>;de)
. . . | | . | . 11. covers_pred(A;I)
. . . | | . | . 12. guarded_trace(A.da;te;I)
. . . | | . | . 13. closed_pred(I)
```

```
. . . | | . | . 14. single_valued_decls(A.ds)
. . . | | . | . 15. s0 : [[A]] rho de e.state
. . . | | . | . 16. x : [[A]] rho de e.state
. . . | | . | . 17. act : [[A]] rho de e.action
. . . | | . | . 18. x' : [[A]] rho de e.state
. . . | | . | . 19. tr : (Σ[[A.da]] rho) List
. . . | | . | . 20. [[A]] rho de e.init(s0)
. . . | | . | . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | | . | . |                                  te).trace;x)
. . . | | . | . 22. ∀r:rel().
. . . | | . | . |   r ∈ I
. . . | | . | . |   ⇒
. . . | | . | . |   [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | . | . 23. [[A]] rho de e.trans(x,act,x')
. . . | | . | . 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | . | . |   ⇒
. . . | | . | . |   (∀r:rel().
. . . | | . | . |    r ∈ I
. . . | | . | . |    ⇒
. . . | | . | . |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | . 25. ...
. . . | | . | . |   ⇒
. . . | | . | . |   (∀r:rel().
. . . | | . | . |    r ∈ I
. . . | | . | . |    ⇒
. . . | | . | . |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | . 26. r : rel()
. . . | | . | . 27. r ∈ I
. . . | | . | . 28. act ∈ (Σ[[A.da]] rho)
. . . | | . | . 29. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | . | . |   ⇒
. . . | | . | . |   [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | . | . 30. r@0 : rel()
. . . | | . | . 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | . | . 32. rel_unprime(r@0) = rel_unprime(r)
. . . | | . | . 33. ¬affects_trace_rel(te;kind(act);r)
. . . | | . | . 34. [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...
. . . | | . | . ⊢  [[rel_unprime(r)]] rho A.ds ... de e x ... ...
. . . | | . | . | by Assert
. . . | | . | . |    [[r]] rho A.ds dec_lookup(...;...) de e x ... ...
. . . | | . | . |
. . . | | . | . |
. . . | | . | . |\
. . . | | . | . | .....assertion .....
. . . | | . | . | 1. A : ioa{i:l}()
. . . | | . | . | 2. I : Collection(rel())
. . . | | . | . | 3. rho : Decl
. . . | | . | . | 4. de : sig()
. . . | | . | . | 5. e : {[[de]] rho}
. . . | | . | . | 6. te : Label→Label→𝔹
. . . | | . | . | 7. tc_ioa(A;de)
. . . | | . | . | 8. ¬ioa_mentions_trace(A)
. . . | | . | . | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | . | . | 10. tc_pred(I;A.ds;<>;de)
. . . | | . | . | 11. covers_pred(A;I)
```

```
. . . | | . | . | 12. guarded_trace(A.da;te;I)
. . . | | . | . | 13. closed_pred(I)
. . . | | . | . | 14. single_valued_decls(A.ds)
. . . | | . | . | 15. s0 : [[A]] rho de e.state
. . . | | . | . | 16. x : [[A]] rho de e.state
. . . | | . | . | 17. act : [[A]] rho de e.action
. . . | | . | . | 18. x' : [[A]] rho de e.state
. . . | | . | . | 19. tr : (Σ[[A.da]] rho) List
. . . | | . | . | 20. [[A]] rho de e.init(s0)
. . . | | . | . | 21. trace_reachable([[A]] rho de e;s0;....trace;x)
. . . | | . | . | 22. ∀r:rel().
. . . | | . | . | |   r ∈ I
. . . | | . | . | |   ⇒
. . . | | . | . | |   [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | . | . | 23. [[A]] rho de e.trans(x,act,x')
. . . | | . | . | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | . | . | |   ⇒
. . . | | . | . | |   (∀r:rel().
. . . | | . | . | |    r ∈ I
. . . | | . | . | |    ⇒
. . . | | . | . | |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | . | 25. (∀r:rel().
. . . | | . | . | |    (∃r@0:rel().
. . . | | . | . | |     r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | . | . | |    ⇒
. . . | | . | . | |    [[r]] rho A.ds ... de e x value(act) ...)
. . . | | . | . | |   ⇒
. . . | | . | . | |   (∀r:rel().
. . . | | . | . | |    r ∈ I
. . . | | . | . | |    ⇒
. . . | | . | . | |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | . | 26. r : rel()
. . . | | . | . | 27. r ∈ I
. . . | | . | . | 28. act ∈ (Σ[[A.da]] rho)
. . . | | . | . | 29. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | . | . | |   ⇒
. . . | | . | . | |   [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | . | . | 30. r@0 : rel()
. . . | | . | . | 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | . | . | 32. rel_unprime(r@0) = rel_unprime(r)
. . . | | . | . | 33. ¬affects_trace_rel(te;kind(act);r)
. . . | | . | . | 34. [[r]] rho A.ds ... de e x value(act) ...
. . . | | . | . | ⊢  [[r]] rho A.ds dec_lookup(...;...) de e x ... ...
. . . | | . | . | | by NthHypSq (-1)
. . . | | . | . | |
. . . | | . | . | .....assertion .....
. . . | | . | . | 1. A : ioa{i:l}()
. . . | | . | . | 2. I : Collection(rel())
. . . | | . | . | 3. rho : Decl
. . . | | . | . | 4. de : sig()
. . . | | . | . | 5. e : {[[de]] rho}
. . . | | . | . | 6. te : Label→Label→𝔹
. . . | | . | . | 7. tc_ioa(A;de)
. . . | | . | . | 8. ¬ioa_mentions_trace(A)
. . . | | . | . | 9. trace_consistent_pred(rho;A.da;te;I)
```

```
. . . | | . | . | 10. tc_pred(I;A.ds;<>;de)
. . . | | . | . | 11. covers_pred(A;I)
. . . | | . | . | 12. guarded_trace(A.da;te;I)
. . . | | . | . | 13. closed_pred(I)
. . . | | . | . | 14. single_valued_decls(A.ds)
. . . | | . | . | 15. s0 : [[A]] rho de e.state
. . . | | . | . | 16. x : [[A]] rho de e.state
. . . | | . | . | 17. act : [[A]] rho de e.action
. . . | | . | . | 18. x' : [[A]] rho de e.state
. . . | | . | . | 19. tr : (Σ[[A.da]] rho) List
. . . | | . | . | 20. [[A]] rho de e.init(s0)
. . . | | . | . | 21. trace_reachable([[A]] rho de e;s0;....trace;x)
. . . | | . | . | 22. ∀r:rel().
. . . | | . | . |     r ∈ I
. . . | | . | . |     ⇒
. . . | | . | . |     [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | . | . | 23. [[A]] rho de e.trans(x,act,x')
. . . | | . | . | 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | . | . |     ⇒
. . . | | . | . |     (∀r:rel().
. . . | | . | . |      r ∈ I
. . . | | . | . |      ⇒
. . . | | . | . |      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | . | 25. (∀r:rel().
. . . | | . | . |      (∃r@0:rel().
. . . | | . | . |       r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | . | . |      ⇒
. . . | | . | . |      [[r]] rho A.ds ... de e x value(act) ...)
. . . | | . | . |     ⇒
. . . | | . | . |     (∀r:rel().
. . . | | . | . |      r ∈ I
. . . | | . | . |      ⇒
. . . | | . | . |      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | . | 26. r : rel()
. . . | | . | . | 27. r ∈ I
. . . | | . | . | 28. act ∈ (Σ[[A.da]] rho)
. . . | | . | . | 29. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | . | . |     ⇒
. . . | | . | . |     [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | . | . | 30. r@0 : rel()
. . . | | . | . | 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | . | . | 32. rel_unprime(r@0) = rel_unprime(r)
. . . | | . | . | 33. ¬affects_trace_rel(te;kind(act);r)
. . . | | . | . | 34. [[r]] rho A.ds ... de e x value(act) ...
. . . | | . | . | ⊢  [[r]] rho A.ds ... de e x value(act) ... ~ ...
. . . | | . | . |   by ...
. . . | | . | . |
. . . | | . | . .\
. . . | | . | . . 1. A : ioa{i:l}()
. . . | | . | . . 2. I : Collection(rel())
. . . | | . | . . 3. rho : Decl
. . . | | . | . . 4. de : sig()
. . . | | . | . . 5. e : {[[de]] rho}
. . . | | . | . . 6. te : Label→Label→𝔹
. . . | | . | . . 7. tc_ioa(A;de)
```

```
. . . | | . | . . 8. ¬ioa_mentions_trace(A)
. . . | | . | . . 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | . | . . 10. tc_pred(I;A.ds;<>;de)
. . . | | . | . . 11. covers_pred(A;I)
. . . | | . | . . 12. guarded_trace(A.da;te;I)
. . . | | . | . . 13. closed_pred(I)
. . . | | . | . . 14. single_valued_decls(A.ds)
. . . | | . | . . 15. s0 : [[A]] rho de e.state
. . . | | . | . . 16. x : [[A]] rho de e.state
. . . | | . | . . 17. act : [[A]] rho de e.action
. . . | | . | . . 18. x' : [[A]] rho de e.state
. . . | | . | . . 19. tr : (Σ[[A.da]] rho) List
. . . | | . | . . 20. [[A]] rho de e.init(s0)
. . . | | . | . . 21. trace_reachable([[A]] rho de e;s0;....trace;x)
. . . | | . | . . 22. ∀r:rel().
. . . | | . | . .     r ∈ I
. . . | | . | . .     ⇒
. . . | | . | . .     [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | . | . . 23. [[A]] rho de e.trans(x,act,x')
. . . | | . | . . 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | . | . .     ⇒
. . . | | . | . .     (∀r:rel().
. . . | | . | . .      r ∈ I
. . . | | . | . .      ⇒
. . . | | . | . .      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | . . 25. (∀r:rel().
. . . | | . | . .      (∃r@0:rel().
. . . | | . | . .       r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | . | . .      ⇒
. . . | | . | . .      [[r]] rho A.ds ... de e x value(act) ...)
. . . | | . | . .     ⇒
. . . | | . | . .     (∀r:rel().
. . . | | . | . .      r ∈ I
. . . | | . | . .      ⇒
. . . | | . | . .      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . | . . 26. r : rel()
. . . | | . | . . 27. r ∈ I
. . . | | . | . . 28. act ∈ (Σ[[A.da]] rho)
. . . | | . | . . 29. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | | . | . .     ⇒
. . . | | . | . .     [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | | . | . . 30. r@0 : rel()
. . . | | . | . . 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | . | . . 32. rel_unprime(r@0) = rel_unprime(r)
. . . | | . | . . 33. ¬affects_trace_rel(te;kind(act);r)
. . . | | . | . . 34. [[r]] rho A.ds ... de e x value(act) ...
. . . | | . | . . 35. [[r]] rho A.ds ... de e x value(act) ...
. . . | | . | . . ⊢  [[rel_unprime(r)]] rho A.ds ... de e x ... ...
. . . | | . | . .   by NthHypRevSq (-1) THEN Easy
. . . | | . |
. . . | | . .\
. . . | | . . 1. A : ioa{i:l}()
. . . | | . . 2. I : Collection(rel())
. . . | | . . 3. rho : Decl
. . . | | . . 4. de : sig()
```

```
. . . | | . . 5. e : {[[de]] rho}
. . . | | . . 6. te : Label→Label→𝔹
. . . | | . . 7. tc_ioa(A;de)
. . . | | . . 8. ¬ioa_mentions_trace(A)
. . . | | . . 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | . . 10. tc_pred(I;A.ds;<>;de)
. . . | | . . 11. covers_pred(A;I)
. . . | | . . 12. guarded_trace(A.da;te;I)
. . . | | . . 13. closed_pred(I)
. . . | | . . 14. single_valued_decls(A.ds)
. . . | | . . 15. s0 : [[A]] rho de e.state
. . . | | . . 16. x : [[A]] rho de e.state
. . . | | . . 17. act : [[A]] rho de e.action
. . . | | . . 18. x' : [[A]] rho de e.state
. . . | | . . 19. tr : (Σ[[A.da]] rho) List
. . . | | . . 20. [[A]] rho de e.init(s0)
. . . | | . . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | | . . |                                    te).trace;x)
. . . | | . . 22. ∀r:rel().
. . . | | . . |   r ∈ I
. . . | | . . |   ⇒
. . . | | . . |   [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | . . 23. [[A]] rho de e.trans(x,act,x')
. . . | | . . 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | . . |   ⇒
. . . | | . . |   (∀r:rel().
. . . | | . . |    r ∈ I
. . . | | . . |    ⇒
. . . | | . . |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . . 25. (∀r:rel().
. . . | | . . |    (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | . . |    ⇒
. . . | | . . |    [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...)
. . . | | . . |   ⇒
. . . | | . . |   (∀r:rel().
. . . | | . . |    r ∈ I
. . . | | . . |    ⇒
. . . | | . . |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . . 26. r : rel()
. . . | | . . 27. r ∈ I
. . . | | . . 28. act ∈ (Σ[[A.da]] rho)
. . . | | . . 29. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | . . |                                      te);act)
. . . | | . . |   ⇒
. . . | | . . |   [[wp_rel(A;kind(act);r)]] rho ... ... de e x ... ...
. . . | | . . 30. r@0 : rel()
. . . | | . . 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | . . 32. rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | | . . 33. ¬affects_trace_rel(te;kind(act);r)
. . . | | . . 34. [[rel_unprime(r@0)]] rho A.ds ... de e x ... ...
. . . | | . . ⊢  [[r@0]] rho A.ds dec_lookup(A.da;...) de e x ... ...
. . . | | . . | by NthHypSq (-1)
. . . | | . . |
. . . | | . . .....assertion .....
. . . | | . . 1. A : ioa{i:l}()
```

```
. . . | | . . 2. I : Collection(rel())
. . . | | . . 3. rho : Decl
. . . | | . . 4. de : sig()
. . . | | . . 5. e : {[[de]] rho}
. . . | | . . 6. te : Label→Label→𝔹
. . . | | . . 7. tc_ioa(A;de)
. . . | | . . 8. ¬ioa_mentions_trace(A)
. . . | | . . 9. trace_consistent_pred(rho;A.da;te;I)
. . . | | . . 10. tc_pred(I;A.ds;<>;de)
. . . | | . . 11. covers_pred(A;I)
. . . | | . . 12. guarded_trace(A.da;te;I)
. . . | | . . 13. closed_pred(I)
. . . | | . . 14. single_valued_decls(A.ds)
. . . | | . . 15. s0 : [[A]] rho de e.state
. . . | | . . 16. x : [[A]] rho de e.state
. . . | | . . 17. act : [[A]] rho de e.action
. . . | | . . 18. x' : [[A]] rho de e.state
. . . | | . . 19. tr : (Σ[[A.da]] rho) List
. . . | | . . 20. [[A]] rho de e.init(s0)
. . . | | . . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | | . .                                   te).trace;x)
. . . | | . . 22. ∀r:rel().
. . . | | . .     r ∈ I
. . . | | . .     ⇒
. . . | | . .     [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | | . . 23. [[A]] rho de e.trans(x,act,x')
. . . | | . . 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | | . .     ⇒
. . . | | . .     (∀r:rel().
. . . | | . .      r ∈ I
. . . | | . .      ⇒
. . . | | . .      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . . 25. (∀r:rel().
. . . | | . .      (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | | . .      ⇒
. . . | | . .      [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...)
. . . | | . .     ⇒
. . . | | . .     (∀r:rel().
. . . | | . .      r ∈ I
. . . | | . .      ⇒
. . . | | . .      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | | . . 26. r : rel()
. . . | | . . 27. r ∈ I
. . . | | . . 28. act ∈ (Σ[[A.da]] rho)
. . . | | . . 29. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | | . .                                     te);act)
. . . | | . .     ⇒
. . . | | . .     [[wp_rel(A;kind(act);r)]] rho ... ... de e x ... ...
. . . | | . . 30. r@0 : rel()
. . . | | . . 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | | . . 32. rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | | . . 33. ¬affects_trace_rel(te;kind(act);r)
. . . | | . . 34. [[rel_unprime(r@0)]] rho A.ds ... de e x ... ...
. . . | | . . ⊢  [[r@0]] rho A.ds ... de e x value(act) ... ∼ ...
. . . | | . .   by BackThruLemma `rel_mng_unprime` THEN Auto
```

```
. . . | |
. . . | .\
. . . | . 1. A : ioa{i:l}()
. . . | . 2. I : Collection(rel())
. . . | . 3. rho : Decl
. . . | . 4. de : sig()
. . . | . 5. e : {[[de]] rho}
. . . | . 6. te : Label→Label→𝔹
. . . | . 7. tc_ioa(A;de)
. . . | . 8. ¬ioa_mentions_trace(A)
. . . | . 9. trace_consistent_pred(rho;A.da;te;I)
. . . | . 10. tc_pred(I;A.ds;<>;de)
. . . | . 11. covers_pred(A;I)
. . . | . 12. guarded_trace(A.da;te;I)
. . . | . 13. closed_pred(I)
. . . | . 14. single_valued_decls(A.ds)
. . . | . 15. s0 : [[A]] rho de e.state
. . . | . 16. x : [[A]] rho de e.state
. . . | . 17. act : [[A]] rho de e.action
. . . | . 18. x' : [[A]] rho de e.state
. . . | . 19. tr : (Σ[[A.da]] rho) List
. . . | . 20. [[A]] rho de e.init(s0)
. . . | . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | . |                                       te).trace;x)
. . . | . 22. ∀r:rel().
. . . | . |   r ∈ I ⇒ [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | . 23. [[A]] rho de e.trans(x,act,x')
. . . | . 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . | . |   ⇒
. . . | . |   (∀r:rel().
. . . | . |    r ∈ I
. . . | . |    ⇒
. . . | . |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | . |                                          te);act))
. . . | . 25. (∀r:rel().
. . . | . |    (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | . |    ⇒
. . . | . |    [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...)
. . . | . |   ⇒
. . . | . |   (∀r:rel().
. . . | . |    r ∈ I
. . . | . |    ⇒
. . . | . |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | . |                                          te);act))
. . . | . 26. r : rel()
. . . | . 27. r ∈ I
. . . | . 28. act ∈ (Σ[[A.da]] rho)
. . . | . 29. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | . |                                         te);act)
. . . | . |   ⇒
. . . | . |   [[wp_rel(A;kind(act);r)]] rho A.ds ... de e x ... ...
. . . | . 30. r@0 : rel()
. . . | . 31. r@0 ∈ wp_rel(A;kind(act);r)
. . . | . 32. ¬rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | . ⊢  [[r@0]] rho A.ds dec_lookup(A.da;...) de e x ... ...
```

```
. . . | . | by Unfold ‘wp_rel‘ (-2) THEN D (-9)
. . . | . |
. . . | . |\
. . . | . | .....antecedent .....
. . . | . | 1. A : ioa{i:l}()
. . . | . | 2. I : Collection(rel())
. . . | . | 3. rho : Decl
. . . | . | 4. de : sig()
. . . | . | 5. e : {[[de]] rho}
. . . | . | 6. te : Label→Label→𝔹
. . . | . | 7. tc_ioa(A;de)
. . . | . | 8. ¬ioa_mentions_trace(A)
. . . | . | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | . | 10. tc_pred(I;A.ds;<>;de)
. . . | . | 11. covers_pred(A;I)
. . . | . | 12. guarded_trace(A.da;te;I)
. . . | . | 13. closed_pred(I)
. . . | . | 14. single_valued_decls(A.ds)
. . . | . | 15. s0 : [[A]] rho de e.state
. . . | . | 16. x : [[A]] rho de e.state
. . . | . | 17. act : [[A]] rho de e.action
. . . | . | 18. x’ : [[A]] rho de e.state
. . . | . | 19. tr : (Σ[[A.da]] rho) List
. . . | . | 20. [[A]] rho de e.init(s0)
. . . | . | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | . |                                       te).trace;x)
. . . | . | 22. ∀r:rel().
. . . | . |     r ∈ I
. . . | . |     ⇒
. . . | . |     [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | . | 23. [[A]] rho de e.trans(x,act,x’)
. . . | . | 24. (∀r:rel().
. . . | . |      (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | . |      ⇒
. . . | . |      [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...)
. . . | . |     ⇒
. . . | . |     (∀r:rel().
. . . | . |      r ∈ I
. . . | . |      ⇒
. . . | . |      [[r]] rho A.ds <> de e x’ · tappend(mk_trace_env(tr,
. . . | . |                                          te);act))
. . . | . | 25. r : rel()
. . . | . | 26. r ∈ I
. . . | . | 27. act ∈ (Σ[[A.da]] rho)
. . . | . | 28. [[r]] rho A.ds <> de e x’ · tappend(mk_trace_env(tr,
. . . | . |                                         te);act)
. . . | . |     ⇒
. . . | . |     [[wp_rel(A;kind(act);r)]] rho A.ds ... de e x ... ...
. . . | . | 29. r@0 : rel()
. . . | . | 30. r@0 ∈ smts_eff_rel(action_effect(kind(act);...;...);r)
. . . | . | 31. ¬rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | . | ⊢  ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
. . . | . |   by InstLemma ‘rel_effect_lemma‘ [A;rho;de;act;r@0;r
. . . | . |      ] THEN Auto
. . . | . |    THEN Try TrivialIoaHyp
```

```
. . . | . |
. . . | . .\
. . . | . . 1. A : ioa{i:l}()
. . . | . . 2. I : Collection(rel())
. . . | . . 3. rho : Decl
. . . | . . 4. de : sig()
. . . | . . 5. e : {[[de]] rho}
. . . | . . 6. te : Label→Label→𝔹
. . . | . . 7. tc_ioa(A;de)
. . . | . . 8. ¬ioa_mentions_trace(A)
. . . | . . 9. trace_consistent_pred(rho;A.da;te;I)
. . . | . . 10. tc_pred(I;A.ds;<>;de)
. . . | . . 11. covers_pred(A;I)
. . . | . . 12. guarded_trace(A.da;te;I)
. . . | . . 13. closed_pred(I)
. . . | . . 14. single_valued_decls(A.ds)
. . . | . . 15. s0 : [[A]] rho de e.state
. . . | . . 16. x : [[A]] rho de e.state
. . . | . . 17. act : [[A]] rho de e.action
. . . | . . 18. x' : [[A]] rho de e.state
. . . | . . 19. tr : (Σ[[A.da]] rho) List
. . . | . . 20. [[A]] rho de e.init(s0)
. . . | . . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | . . |                                     te).trace;x)
. . . | . . 22. ∀r:rel().
. . . | . . |   r ∈ I
. . . | . . |   ⇒
. . . | . . |   [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | . . 23. [[A]] rho de e.trans(x,act,x')
. . . | . . 24. (∀r:rel().
. . . | . . |    (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | . . |    ⇒
. . . | . . |    [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...)
. . . | . . |   ⇒
. . . | . . |   (∀r:rel().
. . . | . . |    r ∈ I
. . . | . . |    ⇒
. . . | . . |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | . . |                                          te);act))
. . . | . . 25. r : rel()
. . . | . . 26. r ∈ I
. . . | . . 27. act ∈ (Σ[[A.da]] rho)
. . . | . . 28. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | . . |                                         te);act)
. . . | . . |   ⇒
. . . | . . |   [[wp_rel(A;kind(act);r)]] rho A.ds ... de e x ... ...
. . . | . . 29. r@0 : rel()
. . . | . . 30. r@0 ∈ smts_eff_rel(action_effect(kind(act);...;...);r)
. . . | . . 31. ¬rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | . . 32. ∀r:rel().
. . . | . . |   r ∈ I
. . . | . . |   ⇒
. . . | . . |   [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | . . |                                         te);act)
. . . | . . ⊢  [[r@0]] rho A.ds dec_lookup(A.da;...) de e x ... ...
```

```
. . . | . . | by InstHyp [r] (-1) THENA Auto
. . . | . . |
. . . | . . 1. A : ioa{i:l}()
. . . | . . 2. I : Collection(rel())
. . . | . . 3. rho : Decl
. . . | . . 4. de : sig()
. . . | . . 5. e : {[[de]] rho}
. . . | . . 6. te : Label→Label→𝔹
. . . | . . 7. tc_ioa(A;de)
. . . | . . 8. ¬ioa_mentions_trace(A)
. . . | . . 9. trace_consistent_pred(rho;A.da;te;I)
. . . | . . 10. tc_pred(I;A.ds;<>;de)
. . . | . . 11. covers_pred(A;I)
. . . | . . 12. guarded_trace(A.da;te;I)
. . . | . . 13. closed_pred(I)
. . . | . . 14. single_valued_decls(A.ds)
. . . | . . 15. s0 : [[A]] rho de e.state
. . . | . . 16. x : [[A]] rho de e.state
. . . | . . 17. act : [[A]] rho de e.action
. . . | . . 18. x' : [[A]] rho de e.state
. . . | . . 19. tr : (Σ[[A.da]] rho) List
. . . | . . 20. [[A]] rho de e.init(s0)
. . . | . . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | . . |                                     te).trace;x)
. . . | . . 22. ∀r:rel().
. . . | . . |   r ∈ I
. . . | . . |   ⇒
. . . | . . |   [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | . . 23. [[A]] rho de e.trans(x,act,x')
. . . | . . 24. (∀r:rel().
. . . | . . |    (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | . . |    ⇒
. . . | . . |    [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...)
. . . | . . |   ⇒
. . . | . . |   (∀r:rel().
. . . | . . |    r ∈ I
. . . | . . |    ⇒
. . . | . . |    [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | . . |                                        te);act))
. . . | . . 25. r : rel()
. . . | . . 26. r ∈ I
. . . | . . 27. act ∈ (Σ[[A.da]] rho)
. . . | . . 28. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | . . |                                       te);act)
. . . | . . |   ⇒
. . . | . . |   [[wp_rel(A;kind(act);r)]] rho A.ds ... de e x ... ...
. . . | . . 29. r@0 : rel()
. . . | . . 30. r@0 ∈ smts_eff_rel(action_effect(kind(act);...;...);r)
. . . | . . 31. ¬rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | . . 32. ∀r:rel().
. . . | . . |   r ∈ I
. . . | . . |   ⇒
. . . | . . |   [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | . . |                                       te);act)
. . . | . . 33. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
```

```
. . . | . . |                                       te);act)
. . . | . . ⊢  [[r@0]] rho A.ds dec_lookup(A.da;...) de e x ... ...
. . . | . . | by Assert
. . . | . . |    [[wp_rel(A;kind(act);r)]] rho A.ds ... de e x ... ...
. . . | . . |
. . . | . . |
. . . | . . |
. . . | . . |\
. . . | . . | .....assertion .....
. . . | . . | 1. A : ioa{i:l}()
. . . | . . | 2. I : Collection(rel())
. . . | . . | 3. rho : Decl
. . . | . . | 4. de : sig()
. . . | . . | 5. e : {[[de]] rho}
. . . | . . | 6. te : Label→Label→𝔹
. . . | . . | 7. tc_ioa(A;de)
. . . | . . | 8. ¬ioa_mentions_trace(A)
. . . | . . | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | . . | 10. tc_pred(I;A.ds;<>;de)
. . . | . . | 11. covers_pred(A;I)
. . . | . . | 12. guarded_trace(A.da;te;I)
. . . | . . | 13. closed_pred(I)
. . . | . . | 14. single_valued_decls(A.ds)
. . . | . . | 15. s0 : [[A]] rho de e.state
. . . | . . | 16. x : [[A]] rho de e.state
. . . | . . | 17. act : [[A]] rho de e.action
. . . | . . | 18. x' : [[A]] rho de e.state
. . . | . . | 19. tr : (Σ[[A.da]] rho) List
. . . | . . | 20. [[A]] rho de e.init(s0)
. . . | . . | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | . . | |                                    te).trace;x)
. . . | . . | 22. ∀r:rel().
. . . | . . | |   r ∈ I
. . . | . . | |   ⇒
. . . | . . | |   [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | . . | 23. [[A]] rho de e.trans(x,act,x')
. . . | . . | 24. (∀r:rel().
. . . | . . | |    (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | . . | |    ⇒
. . . | . . | |    [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...)
. . . | . . | |   ⇒
. . . | . . | |   (∀r:rel().
. . . | . . | |    r ∈ I
. . . | . . | |    ⇒
. . . | . . | |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | . . | 25. r : rel()
. . . | . . | 26. r ∈ I
. . . | . . | 27. act ∈ (Σ[[A.da]] rho)
. . . | . . | 28. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | . . | |                                        te);act)
. . . | . . | |   ⇒
. . . | . . | |   [[wp_rel(A;kind(act);r)]] rho ... ... de e x ... ...
. . . | . . | 29. r@0 : rel()
. . . | . . | 30. r@0 ∈ smts_eff_rel(action_effect(...;A.eff;...);r)
. . . | . . | 31. ¬rel_eq(rel_unprime(r@0);rel_unprime(r))
```

```
. . . | . . | 32. ∀r:rel().
. . . | . . | |   r ∈ I
. . . | . . | |   ⇒
. . . | . . | |   [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | . . | |                                         te);act)
. . . | . . | 33. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | . . | |                                         te);act)
. . . | . . | ⊢  [[wp_rel(A;kind(act);r)]] rho A.ds ... de e x ... ...
. . . | . . | | by  InstLemma `wp_rel_correctness` [A;r;rho;de;e;act;
. . . | . . | |    tappend(mk_trace_env(tr, te);act);x;x'
. . . | . . | |    ] THENA (Auto THEN TrivialIoaHyp)
. . . | . . |
. . . | . . | |    ...
. . . | . . | THEN EasyHypThen Auto
. . . | . . | |
. . . | . . | |\
. . . | . . | | 1. A : ioa{i:l}()
. . . | . . | | 2. I : Collection(rel())
. . . | . . | | 3. rho : Decl
. . . | . . | | 4. de : sig()
. . . | . . | | 5. e : {[[de]] rho}
. . . | . . | | 6. te : Label→Label→𝔹
. . . | . . | | 7. tc_ioa(A;de)
. . . | . . | | 8. ¬ioa_mentions_trace(A)
. . . | . . | | 9. trace_consistent_pred(rho;A.da;te;I)
. . . | . . | | 10. tc_pred(I;A.ds;<>;de)
. . . | . . | | 11. covers_pred(A;I)
. . . | . . | | 12. guarded_trace(A.da;te;I)
. . . | . . | | 13. closed_pred(I)
. . . | . . | | 14. single_valued_decls(A.ds)
. . . | . . | | 15. s0 : [[A]] rho de e.state
. . . | . . | | 16. x : [[A]] rho de e.state
. . . | . . | | 17. act : [[A]] rho de e.action
. . . | . . | | 18. x' : [[A]] rho de e.state
. . . | . . | | 19. tr : (Σ[[A.da]] rho) List
. . . | . . | | 20. [[A]] rho de e.init(s0)
. . . | . . | | 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | . . | |                                         te).trace;x)
. . . | . . | | 22. ∀r:rel().
. . . | . . | |     r ∈ I
. . . | . . | |     ⇒
. . . | . . | |     [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | . . | | 23. [[A]] rho de e.trans(x,act,x')
. . . | . . | | 24. ...
. . . | . . | |     ⇒
. . . | . . | |     (∀r:rel().
. . . | . . | |      r ∈ I
. . . | . . | |      ⇒
. . . | . . | |      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | . . | | 25. r : rel()
. . . | . . | | 26. r ∈ I
. . . | . . | | 27. act ∈ (Σ[[A.da]] rho)
. . . | . . | | 28. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | . . | |     ⇒
. . . | . . | |     [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
```

```
. . . | . . | | 29. r@0 : rel()
. . . | . . | | 30. r@0 ∈ smts_eff_rel(action_effect(...;A.eff;...);r)
. . . | . . | | 31. ¬rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | . . | | 32. ∀r:rel().
. . . | . . | |     r ∈ I
. . . | . . | |     ⇒
. . . | . . | |     [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | . . | | 33. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | . . | | ⊢  covers_rel(A;r)  by Easy
. . . | . . | |
. . . | . . | .\
. . . | . . | . 1. A : ioa{i:l}()
. . . | . . | . 2. I : Collection(rel())
. . . | . . | . 3. rho : Decl
. . . | . . | . 4. de : sig()
. . . | . . | . 5. e : {[[de]] rho}
. . . | . . | . 6. te : Label→Label→𝔹
. . . | . . | . 7. tc_ioa(A;de)
. . . | . . | . 8. ¬ioa_mentions_trace(A)
. . . | . . | . 9. trace_consistent_pred(rho;A.da;te;I)
. . . | . . | . 10. tc_pred(I;A.ds;<>;de)
. . . | . . | . 11. covers_pred(A;I)
. . . | . . | . 12. guarded_trace(A.da;te;I)
. . . | . . | . 13. closed_pred(I)
. . . | . . | . 14. single_valued_decls(A.ds)
. . . | . . | . 15. s0 : [[A]] rho de e.state
. . . | . . | . 16. x : [[A]] rho de e.state
. . . | . . | . 17. act : [[A]] rho de e.action
. . . | . . | . 18. x' : [[A]] rho de e.state
. . . | . . | . 19. tr : (Σ[[A.da]] rho) List
. . . | . . | . 20. [[A]] rho de e.init(s0)
. . . | . . | . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | . . | . |                                     te).trace;x)
. . . | . . | . 22. ∀r:rel().
. . . | . . | . |   r ∈ I
. . . | . . | . |   ⇒
. . . | . . | . |   [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | . . | . 23. [[A]] rho de e.trans(x,act,x')
. . . | . . | . 24. ...
. . . | . . | . |   ⇒
. . . | . . | . |   (∀r:rel().
. . . | . . | . |    r ∈ I
. . . | . . | . |    ⇒
. . . | . . | . |    [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | . . | . 25. r : rel()
. . . | . . | . 26. r ∈ I
. . . | . . | . 27. act ∈ (Σ[[A.da]] rho)
. . . | . . | . 28. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | . . | . |   ⇒
. . . | . . | . |   [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | . . | . 29. r@0 : rel()
. . . | . . | . 30. r@0 ∈ smts_eff_rel(action_effect(...;A.eff;...);r)
. . . | . . | . 31. ¬rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | . . | . 32. ∀r:rel().
. . . | . . | . |   r ∈ I
```

```
. . . | . . . | . |   ⇒
. . . | . . . | . |   [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | . . . | . 33. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | . . . | . 34. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | . . . | . |   ⇒
. . . | . . . | . |   [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | . . . | . 35. [[r]] rho A.ds <> de e x' · ... ⇐ ...
. . . | . . . | . ⊢  [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | . . . | . | by ThinTrivial
. . . | . . . | . |
. . . | . . . | . 1. A : ioa{i:l}()
. . . | . . . | . 2. I : Collection(rel())
. . . | . . . | . 3. rho : Decl
. . . | . . . | . 4. de : sig()
. . . | . . . | . 5. e : {[[de]] rho}
. . . | . . . | . 6. te : Label→Label→𝔹
. . . | . . . | . 7. tc_ioa(A;de)
. . . | . . . | . 8. ¬ioa_mentions_trace(A)
. . . | . . . | . 9. trace_consistent_pred(rho;A.da;te;I)
. . . | . . . | . 10. tc_pred(I;A.ds;<>;de)
. . . | . . . | . 11. covers_pred(A;I)
. . . | . . . | . 12. guarded_trace(A.da;te;I)
. . . | . . . | . 13. closed_pred(I)
. . . | . . . | . 14. single_valued_decls(A.ds)
. . . | . . . | . 15. s0 : [[A]] rho de e.state
. . . | . . . | . 16. x : [[A]] rho de e.state
. . . | . . . | . 17. act : [[A]] rho de e.action
. . . | . . . | . 18. x' : [[A]] rho de e.state
. . . | . . . | . 19. tr : (Σ[[A.da]] rho) List
. . . | . . . | . 20. [[A]] rho de e.init(s0)
. . . | . . . | . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | . . . | .                                     te).trace;x)
. . . | . . . | . 22. ∀r:rel().
. . . | . . . | .     r ∈ I
. . . | . . . | .     ⇒
. . . | . . . | .     [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | . . . | . 23. [[A]] rho de e.trans(x,act,x')
. . . | . . . | . 24. ...
. . . | . . . | .     ⇒
. . . | . . . | .     (∀r:rel().
. . . | . . . | .      r ∈ I
. . . | . . . | .      ⇒
. . . | . . . | .      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | . . . | . 25. r : rel()
. . . | . . . | . 26. r ∈ I
. . . | . . . | . 27. act ∈ (Σ[[A.da]] rho)
. . . | . . . | . 28. r@0 : rel()
. . . | . . . | . 29. r@0 ∈ smts_eff_rel(action_effect(...;A.eff;...);r)
. . . | . . . | . 30. ¬rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | . . . | . 31. ∀r:rel().
. . . | . . . | .     r ∈ I
. . . | . . . | .     ⇒
. . . | . . . | .     [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | . . . | . 32. [[r]] rho A.ds <> de e x' · tappend(...;act)
. . . | . . . | . 33. [[r]] rho A.ds <> de e x' · ... ⇐ ...
```

```
. . . | . . | . 34. [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | . . | . 35. [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | . . | . ⊢  [[wp_rel(A;...;r)]] rho A.ds ... de e x ... ...
. . . | . . | .   by ...
. . . | . . |
. . . | . . .\
. . . | . . . 1. A : ioa{i:l}()
. . . | . . . 2. I : Collection(rel())
. . . | . . . 3. rho : Decl
. . . | . . . 4. de : sig()
. . . | . . . 5. e : {[[de]] rho}
. . . | . . . 6. te : Label→Label→𝔹
. . . | . . . 7. tc_ioa(A;de)
. . . | . . . 8. ¬ioa_mentions_trace(A)
. . . | . . . 9. trace_consistent_pred(rho;A.da;te;I)
. . . | . . . 10. tc_pred(I;A.ds;<>;de)
. . . | . . . 11. covers_pred(A;I)
. . . | . . . 12. guarded_trace(A.da;te;I)
. . . | . . . 13. closed_pred(I)
. . . | . . . 14. single_valued_decls(A.ds)
. . . | . . . 15. s0 : [[A]] rho de e.state
. . . | . . . 16. x : [[A]] rho de e.state
. . . | . . . 17. act : [[A]] rho de e.action
. . . | . . . 18. x' : [[A]] rho de e.state
. . . | . . . 19. tr : (Σ[[A.da]] rho) List
. . . | . . . 20. [[A]] rho de e.init(s0)
. . . | . . . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . | . . .                                    te).trace;x)
. . . | . . . 22. ∀r:rel().
. . . | . . .     r ∈ I
. . . | . . .     ⇒
. . . | . . .     [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . | . . . 23. [[A]] rho de e.trans(x,act,x')
. . . | . . . 24. (∀r:rel().
. . . | . . .      (∃r@0:rel(). r@0 ∈ I & r ∈ wp_rel(A;kind(act);r@0))
. . . | . . .      ⇒
. . . | . . .      [[r]] rho A.ds dec_lookup(A.da;...) de e x ... ...)
. . . | . . .     ⇒
. . . | . . .     (∀r:rel().
. . . | . . .      r ∈ I
. . . | . . .      ⇒
. . . | . . .      [[r]] rho A.ds <> de e x' · tappend(...;act))
. . . | . . . 25. r : rel()
. . . | . . . 26. r ∈ I
. . . | . . . 27. act ∈ (Σ[[A.da]] rho)
. . . | . . . 28. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | . . .                                      te);act)
. . . | . . .     ⇒
. . . | . . .     [[wp_rel(A;kind(act);r)]] rho ... ... de e x ... ...
. . . | . . . 29. r@0 : rel()
. . . | . . . 30. r@0 ∈ smts_eff_rel(action_effect(...;A.eff;...);r)
. . . | . . . 31. ¬rel_eq(rel_unprime(r@0);rel_unprime(r))
. . . | . . . 32. ∀r:rel().
. . . | . . .     r ∈ I
. . . | . . .     ⇒
```

```
. . . | . . .     [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | . . .                                            te);act)
. . . | . . . 33. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . | . . .                                            te);act)
. . . | . . . 34. [[wp_rel(A;kind(act);r)]] rho ... ... de e x ... ...
. . . | . . . ⊢  [[r@0]] rho A.ds dec_lookup(A.da;...) de e x ... ...
. . . | . . .   by ...
. . . |

. . . .\
. . . . 1. A : ioa{i:l}()
. . . . 2. I : Collection(rel())
. . . . 3. rho : Decl
. . . . 4. de : sig()
. . . . 5. e : {[[de]] rho}
. . . . 6. te : Label→Label→𝔹
. . . . 7. tc_ioa(A;de)
. . . . 8. ¬ioa_mentions_trace(A)
. . . . 9. trace_consistent_pred(rho;A.da;te;I)
. . . . 10. tc_pred(I;A.ds;<>;de)
. . . . 11. covers_pred(A;I)
. . . . 12. guarded_trace(A.da;te;I)
. . . . 13. closed_pred(I)
. . . . 14. single_valued_decls(A.ds)
. . . . 15. s0 : [[A]] rho de e.state
. . . . 16. x : [[A]] rho de e.state
. . . . 17. act : [[A]] rho de e.action
. . . . 18. x' : [[A]] rho de e.state
. . . . 19. tr : (Σ[[A.da]] rho) List
. . . . 20. [[A]] rho de e.init(s0)
. . . . 21. trace_reachable([[A]] rho de e;s0;mk_trace_env(tr,
. . . .                                    te).trace;x)
. . . . 22. ∀r:rel().
. . . .     r ∈ I ⇒ [[r]] rho A.ds <> de e x · mk_trace_env(tr, te)
. . . . 23. [[A]] rho de e.trans(x,act,x')
. . . . 24. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
. . . .     ⇒
. . . .     (∀r:rel().
. . . .      r ∈ I
. . . .      ⇒
. . . .      [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . .                                            te);act))
. . . . 25. (∀r:rel().
. . . .      (∃r@0:rel().
. . . .       r@0 ∈ I
. . . .       & r ∈ smts_eff_rel(<e.smt | e ∈ <e ∈ A.eff |
. . . .                                    e.kind =₁ kind(act)>>
. . . .         + <mk_smt(f.var, f.var, f.typ) |
. . . .            f ∈ <f ∈ A.frame | ¬₂kind(act) ∈ f.acts>>;r@0))
. . . .      ⇒
. . . .      [[r]] rho A.ds dec_lookup(A.da;kind(act)) de e x ... ...)
. . . .     ⇒
. . . .     (∀r:rel().
. . . .      r ∈ I
. . . .      ⇒
. . . .      [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
```

```
. . . .                                         te);act))
. . . . 26. r : rel()
. . . . 27. r ∈ I
. . . . 28. act ∈ (Σ[[A.da]] rho)
. . . . 29. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . .                                         te);act)
. . . .     ⇒
. . . .     [[wp_rel(A;kind(act);r)]] rho A.ds ... de e x ... ...
. . . . 30. [[r]] rho A.ds <> de e x' · tappend(mk_trace_env(tr,
. . . .                                         te);act)
. . . . ⊢  [[r]] rho A.ds <> de e x' · mk_trace_env(tr @ [act], te)
. . . .   by Unfold `tappend` (-1) THEN Reduce (-1) THEN Trivial
END
```