```
vc_trace_correctness  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)
|  ⇒
|  guarded_trace(A.da;te;I)
|  ⇒
|  tc_pred(I;A.ds;<>;de)
|  ⇒
|  covers_pred(A;I)
|  ⇒
|  closed_pred(I)
|  ⇒
|  single_valued_decls(A.ds)
|  ⇒
|  let M = [[A]] rho de e in
|  (∀s:M.state, tr:(Σ[[A.da]] rho) List.
|   (M -tr-> s)
|   ⇒
|   [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te))
|  ⇒
|  (M |= always s,tr.[[I]] rho A.ds <> de e s · mk_trace_env(tr, te))
| by Unfold `let` 0 THEN Reduce 0 THEN

|       Auto THEN Try (Unfold `decl` 0 THEN Complete Auto)

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

|       THEN Try TrivialIoaHyp
|  THEN Try Easy•
|
|\
| 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. guarded_trace(A.da;te;I)
| 11. tc_pred(I;A.ds;<>;de)
| 12. covers_pred(A;I)
| 13. closed_pred(I)
| 14. single_valued_decls(A.ds)
| 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| |   ([[A]] rho de e -tr-> s)
| |   ⇒
| |   [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
```

```
| ⊢  ([[A]] rho de e |= always s,tr.[[I]] rho A.ds <> de e s • ...)
| | by BackThruLemma ‘trace_inv_induction‘
|
| |       THENA (Auto THEN Try 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. guarded_trace(A.da;te;I)
| | 11. tc_pred(I;A.ds;<>;de)
| | 12. covers_pred(A;I)
| | 13. closed_pred(I)
| | 14. single_valued_decls(A.ds)
| | 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| | |   ([[A]] rho de e -tr-> s)
| | |   ⇒
| | |   [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| | 16. s : [[A]] rho de e.state
| | 17. tr : [[A]] rho de e.action List
| | ⊢  [[I]] rho A.ds <> de e s • mk_trace_env(tr, te) ∈ Prop
| | | by Assert mk_trace_env(tr, te) ∈ trace_env([[A.da]] rho)
| | |     THEN Auto
| | THEN Reduce 0 THEN Auto
| |
| | |    ...
| | |
| | 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. guarded_trace(A.da;te;I)
| | 11. tc_pred(I;A.ds;<>;de)
| | 12. covers_pred(A;I)
| | 13. closed_pred(I)
| | 14. single_valued_decls(A.ds)
| | 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| |     ([[A]] rho de e -tr-> s)
| |     ⇒
| |     [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| | 16. [[A]] rho de e.state
```

```
| | 17. tr : [[A]] rho de e.action List
| | 18. mk_trace_env(tr, te) ∈ trace_env([[A.da]] rho)
| | ⊢ · ∈ [[<>]] rho  by BackThruLemma ‘empty_sts_mng‘ THEN Auto
| |
| |\
| | 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. guarded_trace(A.da;te;I)
| | 11. tc_pred(I;A.ds;<>;de)
| | 12. covers_pred(A;I)
| | 13. closed_pred(I)
| | 14. single_valued_decls(A.ds)
| | 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| | |   ([[A]] rho de e -tr-> s)
| | |   ⇒
| | |   [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| | ⊢ ∀x:[[A]] rho de e.state.
| | |  [[A]] rho de e.init(x)
| | |  ⇒
| | |  [[I]] rho A.ds <> de e x · mk_trace_env(nil, te)  by Auto
| | |
| | 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. guarded_trace(A.da;te;I)
| | 11. tc_pred(I;A.ds;<>;de)
| | 12. covers_pred(A;I)
| | 13. closed_pred(I)
| | 14. single_valued_decls(A.ds)
| | 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| | |   ([[A]] rho de e -tr-> s)
| | |   ⇒
| | |   [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| | 16. x : [[A]] rho de e.state
| | 17. [[A]] rho de e.init(x)
| | ⊢ [[I]] rho A.ds <> de e x · mk_trace_env(nil, te)
| | | by AllHyps (InstHyp [x;nil]) THENA Auto
| | |
| | |    THEN TrivialIoaHyp THEN Try Easy
| | |
| | |\
| | | 1. A : ioa{i:l}()
```

```
| | | 2. I : Fmla
| | | 3. rho : Decl
| | | 4. de : sig()
| | | 5. e : {[[de]] rho}
| | | 6. te : Label→Label→ℙ
| | | 7. tc_ioa(A;de)
| | | 8. ¬ioa_mentions_trace(A)
| | | 9. trace_consistent_pred(rho;A.da;te;I)
| | | 10. guarded_trace(A.da;te;I)
| | | 11. tc_pred(I;A.ds;<>;de)
| | | 12. covers_pred(A;I)
| | | 13. closed_pred(I)
| | | 14. single_valued_decls(A.ds)
| | | 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| | | |   ([[A]] rho de e -tr-> s)
| | | |   ⇒
| | | |   [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| | | 16. x : [[A]] rho de e.state
| | | 17. [[A]] rho de e.init(x)
| | | ⊢  ([[A]] rho de e -nil-> x)
| | | | by Unfold ‘reachable_via‘ 0 THEN AutoInstConcl [] THEN Auto
| | | |
| | | 1. A : ioa{i:l}()
| | | 2. I : Fmla
| | | 3. rho : Decl
| | | 4. de : sig()
| | | 5. e : {[[de]] rho}
| | | 6. te : Label→Label→ℙ
| | | 7. tc_ioa(A;de)
| | | 8. ¬ioa_mentions_trace(A)
| | | 9. trace_consistent_pred(rho;A.da;te;I)
| | | 10. guarded_trace(A.da;te;I)
| | | 11. tc_pred(I;A.ds;<>;de)
| | | 12. covers_pred(A;I)
| | | 13. closed_pred(I)
| | | 14. single_valued_decls(A.ds)
| | | 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| | |     ([[A]] rho de e -tr-> s)
| | |     ⇒
| | |     [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| | | 16. x : [[A]] rho de e.state
| | | 17. [[A]] rho de e.init(x)
| | | ⊢  trace_reachable([[A]] rho de e;x;nil;x)
| | |   by RecUnfold ‘trace_reachable‘ 0 THEN Reduce 0 THEN Auto
| | |
| | .\
| | . 1. A : ioa{i:l}()
| | . 2. I : Fmla
| | . 3. rho : Decl
| | . 4. de : sig()
| | . 5. e : {[[de]] rho}
| | . 6. te : Label→Label→ℙ
| | . 7. tc_ioa(A;de)
| | . 8. ¬ioa_mentions_trace(A)
| | . 9. trace_consistent_pred(rho;A.da;te;I)
```

```
| | . 10. guarded_trace(A.da;te;I)
| | . 11. tc_pred(I;A.ds;<>;de)
| | . 12. covers_pred(A;I)
| | . 13. closed_pred(I)
| | . 14. single_valued_decls(A.ds)
| | . 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| | . |   ([[A]] rho de e -tr-> s)
| | . |   ⇒
| | . |   [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| | . 16. x : [[A]] rho de e.state
| | . 17. [[A]] rho de e.init(x)
| | . 18. [[VCs(A;I)]] rho A.ds A.da de e x mk_trace_env(nil, te)
| | . ⊢  [[I]] rho A.ds <> de e x · mk_trace_env(nil, te)
| | . | by Unfold `ioa_inv_vc` (-1)
| | . |
| | . |
| | . 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. guarded_trace(A.da;te;I)
| | . 11. tc_pred(I;A.ds;<>;de)
| | . 12. covers_pred(A;I)
| | . 13. closed_pred(I)
| | . 14. single_valued_decls(A.ds)
| | . 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| | . |   ([[A]] rho de e -tr-> s)
| | . |   ⇒
| | . |   [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| | . 16. x : [[A]] rho de e.state
| | . 17. [[A]] rho de e.init(x)
| | . 18. [[<*vc_imp(mk_imp(A.init, I))*>
| | . |   +* ioa_trans_all{i}(A;I)]] rho A.ds A.da de e x ...
| | . ⊢  [[I]] rho A.ds <> de e x · mk_trace_env(nil, te)
| | . | by Unfold  `vcs_mng` (-1) THEN InstHyp [vc_imp(mk_imp(A.init,
| | . |                                                 I))] (-1)
| | .
| | . |    THENA (Auto THEN Unfolds ``vcs_add vcs_singleton`` 0
| | .
| | . |                THEN RW ColMemberC 0
| | .
| | . |                THEN SimpConcl)
| | . |
| | . |
| | . ...
| | . |
| | . 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. guarded_trace(A.da;te;I)
| | . 11. tc_pred(I;A.ds;<>;de)
| | . 12. covers_pred(A;I)
| | . 13. closed_pred(I)
| | . 14. single_valued_decls(A.ds)
| | . 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| | . |   ([[A]] rho de e -tr-> s)
| | . |   ⇒
| | . |   [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| | . 16. x : [[A]] rho de e.state
| | . 17. [[A]] rho de e.init(x)
| | . 18. ∀v:vc{i:l}().
| | . |   v ∈ <*vc_imp(mk_imp(A.init, I))*> +* ioa_trans_all{i}(A;I)
| | . |   ⇒
| | . |   vc_mng(v;rho;A.ds;A.da;de;e;x;mk_trace_env(nil, te))
| | . 19. [[A.init]] rho A.ds <> de e x · mk_trace_env(nil, te)
| | . |   ⇒
| | . |   [[I]] rho A.ds <> de e x · mk_trace_env(nil, te)
| | . ⊢  [[A.init]] rho A.ds <> de e x · mk_trace_env(nil, te)
| | . | by Unfold `ioa_mng` (-3) THEN Reduce (-3)
| | .
| | . |    THEN RepeatFor 2 (ParallelOp (-3))
| | . THEN ParallelOp (-1)
| | . |
| | . 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. guarded_trace(A.da;te;I)
| | . 11. tc_pred(I;A.ds;<>;de)
| | . 12. covers_pred(A;I)
| | . 13. closed_pred(I)
| | . 14. single_valued_decls(A.ds)
| | . 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| | .     ([[A]] rho de e -tr-> s)
| | .     ⇒
| | .     [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| | . 16. x : [[A]] rho de e.state
| | . 17. ∀r:rel().
| | .     r ∈ A.init ⇒ [[r]] rho A.ds <> de e x · niltrace()
| | . 18. ∀v:vc{i:l}().
| | .     v ∈ <*vc_imp(mk_imp(A.init, I))*> +* ioa_trans_all{i}(A;I)
| | .     ⇒
| | .     vc_mng(v;rho;A.ds;A.da;de;e;x;mk_trace_env(nil, te))
| | . 19. [[A.init]] rho A.ds <> de e x · mk_trace_env(nil, te)
```

```
| | .     ⇒
| | .     [[I]] rho A.ds <> de e x · mk_trace_env(nil, te)
| | . 20. r : rel()
| | . 21. r ∈ A.init
| | . 22. [[r]] rho A.ds <> de e x · niltrace()
| | . ⊢  [[r]] rho A.ds <> de e x · mk_trace_env(nil, te)
| | .   by Subst
| | .      [[r]] rho A.ds <> de e x · mk_trace_env(nil,
| | .                                  te) ∼ ...
| | .
| | .      ...
| |
| .\
| . 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. guarded_trace(A.da;te;I)
| . 11. tc_pred(I;A.ds;<>;de)
| . 12. covers_pred(A;I)
| . 13. closed_pred(I)
| . 14. single_valued_decls(A.ds)
| . 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| . |   ([[A]] rho de e -tr-> s)
| . |   ⇒
| . |   [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| . ⊢  ∀s0,x:[[A]] rho de e.state, act:[[A]] rho de e.action
| . |  , x':[[A]] rho de e.state, l:[[A]] rho de e.action List.
| . |  [[A]] rho de e.init(s0)
| . |  ⇒
| . |  trace_reachable([[A]] rho de e;s0;l;x)
| . |  ⇒
| . |  [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . |  ⇒
| . |  [[A]] rho de e.trans(x,act,x')
| . |  ⇒
| . |  [[I]] rho A.ds <> de e x' · mk_trace_env(l @ [act], te)
| . | by Repeat (D 0 THENA Complete Auto)
| . THEN Assert
| . |    mk_trace_env(l, te) ∈ trace_env([[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. guarded_trace(A.da;te;I)
| . | 11. tc_pred(I;A.ds;<>;de)
| . | 12. covers_pred(A;I)
| . | 13. closed_pred(I)
| . | 14. single_valued_decls(A.ds)
| . | 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| . |     ([[A]] rho de e -tr-> s)
| . |     ⇒
| . |     [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| . | 16. s0 : [[A]] rho de e.state
| . | 17. x : [[A]] rho de e.state
| . | 18. [[A]] rho de e.action
| . | 19. [[A]] rho de e.state
| . | 20. l : [[A]] rho de e.action List
| . | 21. [[A]] rho de e.init(s0)
| . | 22. trace_reachable([[A]] rho de e;s0;l;x)
| . | ⊢  mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . |   by Auto THEN Unfold `ioa_mng` (-1) THEN Reduce (-1) THEN Auto
| . |
| . .\
| . . 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. guarded_trace(A.da;te;I)
| . . 11. tc_pred(I;A.ds;<>;de)
| . . 12. covers_pred(A;I)
| . . 13. closed_pred(I)
| . . 14. single_valued_decls(A.ds)
| . . 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| . . |   ([[A]] rho de e -tr-> s)
| . . |   ⇒
| . . |   [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| . . 16. s0 : [[A]] rho de e.state
| . . 17. x : [[A]] rho de e.state
| . . 18. act : [[A]] rho de e.action
| . . 19. x' : [[A]] rho de e.state
| . . 20. l : [[A]] rho de e.action List
| . . 21. [[A]] rho de e.init(s0)
| . . 22. trace_reachable([[A]] rho de e;s0;l;x)
| . . 23. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . ⊢  [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . |  ⇒
| . . |  [[A]] rho de e.trans(x,act,x')
| . . |  ⇒
| . . |  [[I]] rho A.ds <> de e x' · mk_trace_env(l @ [act], te)
| . . | by Auto THEN Reduce 0 THEN Auto
| . .
```

```
| . . |                  THEN 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. guarded_trace(A.da;te;I)
| . . | 11. tc_pred(I;A.ds;<>;de)
| . . | 12. covers_pred(A;I)
| . . | 13. closed_pred(I)
| . . | 14. single_valued_decls(A.ds)
| . . | 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| . . | |   ([[A]] rho de e -tr-> s)
| . . | |   ⇒
| . . | |   [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| . . | 16. s0 : [[A]] rho de e.state
| . . | 17. x : [[A]] rho de e.state
| . . | 18. act : [[A]] rho de e.action
| . . | 19. x' : [[A]] rho de e.state
| . . | 20. l : [[A]] rho de e.action List
| . . | 21. [[A]] rho de e.init(s0)
| . . | 22. trace_reachable([[A]] rho de e;s0;l;x)
| . . | 23. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | 24. [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . | 25. [[A]] rho de e.trans(x,act,x')
| . . | ⊢  [[I]] rho A.ds <> de e x' · mk_trace_env(l @ [act], te)
| . . | | by Assert l ∈ (Σ[[A.da]] rho) List
| . . | |
| . . | |\
| . . | | .....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. guarded_trace(A.da;te;I)
| . . | | 11. tc_pred(I;A.ds;<>;de)
| . . | | 12. covers_pred(A;I)
| . . | | 13. closed_pred(I)
| . . | | 14. single_valued_decls(A.ds)
| . . | | 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| . . | |     ([[A]] rho de e -tr-> s)
| . . | |     ⇒
| . . | |     [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| . . | | 16. s0 : [[A]] rho de e.state
```

```
| . . | | 17. x : [[A]] rho de e.state
| . . | | 18. act : [[A]] rho de e.action
| . . | | 19. x' : [[A]] rho de e.state
| . . | | 20. l : [[A]] rho de e.action List
| . . | | 21. [[A]] rho de e.init(s0)
| . . | | 22. trace_reachable([[A]] rho de e;s0;l;x)
| . . | | 23. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | | 24. [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . | | 25. [[A]] rho de e.trans(x,act,x')
| . . | | ⊢  l ∈ (Σ[[A.da]] rho) List
| . . | |   by ...
| . . | |
| . . | .\
| . . | . 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. guarded_trace(A.da;te;I)
| . . | . 11. tc_pred(I;A.ds;<>;de)
| . . | . 12. covers_pred(A;I)
| . . | . 13. closed_pred(I)
| . . | . 14. single_valued_decls(A.ds)
| . . | . 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| . . | . |   ([[A]] rho de e -tr-> s)
| . . | . |   ⇒
| . . | . |   [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| . . | . 16. s0 : [[A]] rho de e.state
| . . | . 17. x : [[A]] rho de e.state
| . . | . 18. act : [[A]] rho de e.action
| . . | . 19. x' : [[A]] rho de e.state
| . . | . 20. l : [[A]] rho de e.action List
| . . | . 21. [[A]] rho de e.init(s0)
| . . | . 22. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . 23. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | . 24. [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . | . 25. [[A]] rho de e.trans(x,act,x')
| . . | . 26. l ∈ (Σ[[A.da]] rho) List
| . . | . ⊢  [[I]] rho A.ds <> de e x' · mk_trace_env(l @ [act], te)
| . . | . | by AllHyps (\i. InstHyp [x;l] i THENM Thin i ) THENA Auto
| . . | . |
| . . | . |\
| . . | . | .....antecedent .....
| . . | . | 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. guarded_trace(A.da;te;I)
| . . | . | 11. tc_pred(I;A.ds;<>;de)
| . . | . | 12. covers_pred(A;I)
| . . | . | 13. closed_pred(I)
| . . | . | 14. single_valued_decls(A.ds)
| . . | . | 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| . . | . |     ([[A]] rho de e -tr-> s)
| . . | . |     ⇒
| . . | . |     [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| . . | . | 16. s0 : [[A]] rho de e.state
| . . | . | 17. x : [[A]] rho de e.state
| . . | . | 18. act : [[A]] rho de e.action
| . . | . | 19. x' : [[A]] rho de e.state
| . . | . | 20. l : [[A]] rho de e.action List
| . . | . | 21. [[A]] rho de e.init(s0)
| . . | . | 22. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . | 23. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | . | 24. [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . | . | 25. [[A]] rho de e.trans(x,act,x')
| . . | . | 26. l ∈ (Σ[[A.da]] rho) List
| . . | . | ⊢  ([[A]] rho de e -l-> x)
| . . | . |   by ...
| . . | . |
| . . | . .\
| . . | . . 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. guarded_trace(A.da;te;I)
| . . | . . 11. tc_pred(I;A.ds;<>;de)
| . . | . . 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . 20. [[A]] rho de e.init(s0)
| . . | . . 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . 22. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | . . 23. [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . | . . 24. [[A]] rho de e.trans(x,act,x')
| . . | . . 25. l ∈ (Σ[[A.da]] rho) List
| . . | . . 26. [[VCs(A;I)]] rho A.ds A.da de e x mk_trace_env(l, te)
| . . | . . ⊢  [[I]] rho A.ds <> de e x' · mk_trace_env(l @ [act], te)
| . . | . . | 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. guarded_trace(A.da;te;I)
| . . | . . | 11. tc_pred(I;A.ds;<>;de)
| . . | . . | 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . | 20. [[A]] rho de e.init(s0)
| . . | . . | 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . | 22. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | . . | 23. [[I]] rho A.ds <> de e x • mk_trace_env(l, te)
| . . | . . | 24. [[A]] rho de e.trans(x,act,x')
| . . | . . | 25. l ∈ (Σ[[A.da]] rho) List
| . . | . . | 26. [[VCs(A;I)]] rho A.ds A.da de e x mk_trace_env(l,
| . . | . . |                                          te)
| . . | . . | ⊢  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. guarded_trace(A.da;te;I)
| . . | . . 11. tc_pred(I;A.ds;<>;de)
| . . | . . 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . 20. [[A]] rho de e.init(s0)
| . . | . . 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . 22. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | . . 23. [[I]] rho A.ds <> de e x • mk_trace_env(l, te)
| . . | . . 24. [[A]] rho de e.trans(x,act,x')
```

```
| . . | . . . 25. l ∈ (Σ[[A.da]] rho) List
| . . | . . . 26. [[VCs(A;I)]] rho A.ds A.da de e x mk_trace_env(l,
| . . | . . . |                                          te)
| . . | . . . 27. act ∈ (Σ[[A.da]] rho)
| . . | . . . ⊢  [[I]] rho A.ds <> de e x' · mk_trace_env(l @ [act],
| . . | . . . |                                  te)
| . . | . . . | by Assert (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
| . . | . . . |           ⇒
| . . | . . . |    [[I]] rho A.ds <> de e x' · mk_trace_env(l @ [act],
| . . | . . . |                                   te)
| . . | . . . |
| . . | . . . |\
| . . | . . . | .....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. guarded_trace(A.da;te;I)
| . . | . . . | 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | 20. [[A]] rho de e.init(s0)
| . . | . . . | 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | 22. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | . . . | 23. [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . | . . . | 24. [[A]] rho de e.trans(x,act,x')
| . . | . . . | 25. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | 26. [[VCs(A;I)]] rho A.ds A.da de e x mk_trace_env(l,
| . . | . . . | |                                          te)
| . . | . . . | 27. act ∈ (Σ[[A.da]] rho)
| . . | . . . | ⊢  (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
| . . | . . . | |  ⇒
| . . | . . . | |  [[I]] rho A.ds <> de e x' · mk_trace_env(l @ [act],
| . . | . . . | |                                  te)
| . . | . . . | | by D 0  THENA Auto
| . . | . . . | |
| . . | . . . | 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. guarded_trace(A.da;te;I)
| . . | . . . | 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | 20. [[A]] rho de e.init(s0)
| . . | . . . | 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | 22. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | . . . | 23. [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . | . . . | 24. [[A]] rho de e.trans(x,act,x')
| . . | . . . | 25. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | 26. [[VCs(A;I)]] rho A.ds A.da de e x mk_trace_env(l,
| . . | . . . | |                                        te)
| . . | . . . | 27. act ∈ (Σ[[A.da]] rho)
| . . | . . . | 28. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | ⊢  [[I]] rho A.ds <> de e x' · mk_trace_env(l @ [act],
| . . | . . . | |                                 te)
| . . | . . . | | by MoveToHyp (-6) (-1)
| . . | . . . | |
| . . | . . . | 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. guarded_trace(A.da;te;I)
| . . | . . . | 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | 20. [[A]] rho de e.init(s0)
| . . | . . . | 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | 22. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | . . . | 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | 24. [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . | . . . | 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | 27. [[VCs(A;I)]] rho A.ds A.da de e x mk_trace_env(l,
| . . | . . . | |                                        te)
| . . | . . . | 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | ⊢  [[I]] rho A.ds <> de e x' · mk_trace_env(l @ [act],
```

```
| . . | . . . | |                                  te)  by RepD
| . . | . . . | |
| . . | . . . | 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. guarded_trace(A.da;te;I)
| . . | . . . | 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | 20. [[A]] rho de e.init(s0)
| . . | . . . | 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | 22. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | . . . | 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | 24. [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . | . . . | 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | 27. [[VCs(A;I)]] rho A.ds A.da de e x mk_trace_env(l,
| . . | . . . | |                                      te)
| . . | . . . | 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | ⊢  [[I]] rho A.ds <> de e x' · mk_trace_env(l @ [act],
| . . | . . . | |                                 te)
| . . | . . . | | by InstLemma `wp_correctness` [A;de;I;rho;e;act;
| . . | . . . | |    tappend(mk_trace_env(l, te);act);x;x']
| . . | . . . |
| . . | . . . | |    THENA (Auto THEN 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. guarded_trace(A.da;te;I)
| . . | . . . | 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | 20. [[A]] rho de e.init(s0)
| . . | . . . | 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | 22. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | . . . | 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)

| . . | . . . | 24. [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . | . . . | 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | 27. [[VCs(A;I)]] rho A.ds A.da de e x mk_trace_env(l,
| . . | . . . | |                                          te)
| . . | . . . | 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | 29. [[I]] rho A.ds <> de e x' · tappend(...;act)
| . . | . . . | |   ⟺
| . . | . . . | |   [[wp(A;kind(act);I)]] rho A.ds ... de e x ... ...
| . . | . . . | ⊢  [[I]] rho A.ds <> de e x' · mk_trace_env(l @ [act],
| . . | . . . | |                                te)
| . . | . . . | | by D (-1) THEN D (-1)
| . . | . . . | |
| . . | . . . | |\
| . . | . . . | | .....antecedent .....
| . . | . . . | | 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. guarded_trace(A.da;te;I)
| . . | . . . | | 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | 20. [[A]] rho de e.init(s0)
| . . | . . . | | 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | 22. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | . . . | | 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | | 24. [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . | . . . | | 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | | 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | 27. [[VCs(A;I)]] rho A.ds A.da de e x ...
| . . | . . . | | 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | 29. [[I]] rho A.ds <> de e x' · tappend(...;act)
| . . | . . . | | |   ⇒
| . . | . . . | | |   [[wp(A;kind(act);I)]] rho ... ... de e x ... ...
| . . | . . . | | ⊢  [[wp(A;kind(act);I)]] rho A.ds ... de e x ... ...
```

```
| . . | . . . | | | by Thin (-1)
| . . | . . . | | |
| . . | . . . | | 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. guarded_trace(A.da;te;I)
| . . | . . . | | 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | 20. [[A]] rho de e.init(s0)
| . . | . . . | | 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | 22. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | . . . | | 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | | 24. [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . | . . . | | 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | | 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | 27. [[VCs(A;I)]] rho A.ds A.da de e x ...
| . . | . . . | | 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | ⊢  [[wp(A;kind(act);I)]] rho A.ds ... de e x ... ...
| . . | . . . | | | by ...
| . . | . . . | | |
| . . | . . . | | |    ...
| . . | . . . | | THEN InstHyp [ioa_trans(A;kind(act);I)
| . . | . . . | | |    ] (-2) THENA Auto
| . . | . . . | | |
| . . | . . . | | |\
| . . | . . . | | | .....antecedent .....
| . . | . . . | | | 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. guarded_trace(A.da;te;I)
| . . | . . . | | | 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | | 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | | 20. [[A]] rho de e.init(s0)
| . . | . . . | | | 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | | 22. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | . . . | | | 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | | | 24. [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . | . . . | | | 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | | | 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | | 27. ∀v:vc{i:l}().
| . . | . . . | | | |   v ∈ <*vc_imp(mk_imp(A.init, I))*>
| . . | . . . | | | |   ∨ (∃a:dec().
| . . | . . . | | | |      a ∈ A.da & v = ioa_trans(A;a.lbl;I))
| . . | . . . | | | |   ⇒
| . . | . . . | | | |   vc_mng(v;rho;A.ds;A.da;de;e;x;mk_trace_env(l,
| . . | . . . | | | |                                    te))
| . . | . . . | | | 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | | ⊢  ioa_trans(A;kind(act);I) ∈ <*vc_imp(mk_imp(...,
| . . | . . . | | | |                                      I))*>
| . . | . . . | | | |  ∨ (∃a:dec().
| . . | . . . | | | |     a ∈ A.da
| . . | . . . | | | |     & ioa_trans(A;kind(act);I)
| . . | . . . | | | |       =
| . . | . . . | | | |       ioa_trans(A;a.lbl;I))
| . . | . . . | | | | by ...
| . . | . . . | | | |
| . . | . . . | | | 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. guarded_trace(A.da;te;I)
| . . | . . . | | | 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | | 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | | 20. [[A]] rho de e.init(s0)
| . . | . . . | | | 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | | 22. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | . . . | | | 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | | | 24. [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . | . . . | | | 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | | | 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | | 27. ∀v:vc{i:l}().
| . . | . . . | | |     v ∈ <*vc_imp(mk_imp(A.init, I))*>
```

```
| . . | . . . | | |     ∨ (∃a:dec().
| . . | . . . | | |        a ∈ A.da & v = ioa_trans(A;a.lbl;I))
| . . | . . . | | |     ⇒
| . . | . . . | | |     vc_mng(v;rho;A.ds;A.da;de;e;x;mk_trace_env(l,
| . . | . . . | | |                                    te))
| . . | . . . | | | 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | | ⊢  ∃a:dec().
| . . | . . . | | |    a ∈ A.da
| . . | . . . | | |    & ioa_trans(A;kind(act);I)
| . . | . . . | | |      =
| . . | . . . | | |      ioa_trans(A;a.lbl;I)
| . . | . . . | | |   by ...
| . . | . . . | | |
| . . | . . . | | .\
| . . | . . . | | . 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. guarded_trace(A.da;te;I)
| . . | . . . | | . 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | . 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | . 20. [[A]] rho de e.init(s0)
| . . | . . . | | . 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | . 22. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | . . . | | . 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | | . 24. [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . | . . . | | . 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | | . 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | . 27. ∀v:vc{i:l}().
| . . | . . . | | . |   v ∈ <*vc_imp(mk_imp(A.init, I))*>
| . . | . . . | | . |   ∨ (∃a:dec().
| . . | . . . | | . |      a ∈ A.da & v = ioa_trans(A;a.lbl;I))
| . . | . . . | | . |   ⇒
| . . | . . . | | . |   vc_mng(v;rho;A.ds;A.da;de;e;x;mk_trace_env(l,
| . . | . . . | | . |                                    te))
| . . | . . . | | . 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | . 29. vc_mng(...;rho;A.ds;A.da;de;e;x;...)
| . . | . . . | | . ⊢  [[wp(A;...;I)]] rho A.ds ... de e x ... ...
| . . | . . . | | . | by ...
| . . | . . . | | . THEN InstHyp [value(act)] (-1)
| . . | . . . | | . |
| . . | . . . | | . |\
| . . | . . . | | . | .....wf .....
```

```
| . . | . . . | | . | 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. guarded_trace(A.da;te;I)
| . . | . . . | | . | 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | . | 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | . | 20. [[A]] rho de e.init(s0)
| . . | . . . | | . | 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | . | 22. mk_trace_env(l, te)
| . . | . . . | | . |     ∈ trace_env([[A.da]] rho)
| . . | . . . | | . | 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | | . | 24. [[I]] rho A.ds <> de e x · mk_trace_env(l,
| . . | . . . | | . |                                te)
| . . | . . . | | . | 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | | . | 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | . | 27. ∀v:vc{i:l}().
| . . | . . . | | . |     v ∈ <*vc_imp(mk_imp(A.init, I))*>
| . . | . . . | | . |     ∨ (∃a:dec().
| . . | . . . | | . |        a ∈ A.da & v = ioa_trans(A;a.lbl;I))
| . . | . . . | | . |     ⇒
| . . | . . . | | . |     vc_mng(v;rho;A.ds;A.da;de;e;x;...)
| . . | . . . | | . | 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | . | 29. ∀v:[[dec_lookup(A.da;kind(act))]] rho.
| . . | . . . | | . |     [[...]] rho A.ds ... de e x v ...
| . . | . . . | | . |     ⇒
| . . | . . . | | . |     [[wp(A;...;I)]] rho A.ds ... de e x v ...
| . . | . . . | | . | ⊢  value(act)
| . . | . . . | | . |    ∈ [[dec_lookup(A.da;kind(act))]] rho
| . . | . . . | | . |   by Easy
| . . | . . . | | . |
| . . | . . . | | . |\
| . . | . . . | | . | .....antecedent .....
| . . | . . . | | . | 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. guarded_trace(A.da;te;I)
```

```
| . . | . . . | | . | 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | . | 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | . | 20. [[A]] rho de e.init(s0)
| . . | . . . | | . | 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | . | 22. mk_trace_env(l, te)
| . . | . . . | | . | |   ∈ trace_env([[A.da]] rho)
| . . | . . . | | . | 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | | . | 24. [[I]] rho A.ds <> de e x · mk_trace_env(l,
| . . | . . . | | . | |                              te)
| . . | . . . | | . | 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | | . | 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | . | 27. ∀v:vc{i:l}().
| . . | . . . | | . | |   v ∈ <*vc_imp(mk_imp(A.init, I))*>
| . . | . . . | | . | |   ∨ (∃a:dec().
| . . | . . . | | . | |      a ∈ A.da & v = ioa_trans(A;a.lbl;I))
| . . | . . . | | . | |   ⇒
| . . | . . . | | . | |   vc_mng(v;rho;A.ds;A.da;de;e;x;...)
| . . | . . . | | . | 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | . | 29. ∀v:[[dec_lookup(A.da;kind(act))]] rho.
| . . | . . . | | . | |   [[...]] rho A.ds ... de e x v ...
| . . | . . . | | . | |   ⇒
| . . | . . . | | . | |   [[wp(A;...;I)]] rho A.ds ... de e x v ...
| . . | . . . | | . | ⊢  [[...]] rho A.ds ... de e x value(act) ...
| . . | . . . | | . | | by ...
| . . | . . . | | . | THEN Auto
| . . | . . . | | . | |

| . . | . . . | | . | |\
| . . | . . . | | . | | 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. guarded_trace(A.da;te;I)
| . . | . . . | | . | | 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | . | | 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | . | | 20. [[A]] rho de e.init(s0)
| . . | . . . | | . | | 21. trace_reachable([[A]] rho de e;s0;l;x)
```

```
| . . | . . . | | . | | 22. mk_trace_env(l, te)
| . . | . . . | | . | | |   ∈ trace_env([[A.da]] rho)
| . . | . . . | | . | | 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | | . | | 24. [[I]] rho A.ds <> de e x • mk_trace_env(l,
| . . | . . . | | . | | |                              te)
| . . | . . . | | . | | 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | | . | | 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | . | | 27. ∀v:vc{i:l}().
| . . | . . . | | . | | |   v ∈ <*vc_imp(mk_imp(A.init, I))*>
| . . | . . . | | . | | |   ∨ (∃a:dec().
| . . | . . . | | . | | |      a ∈ A.da & v = ioa_trans(A;a.lbl;I))
| . . | . . . | | . | | |   ⇒
| . . | . . . | | . | | |   vc_mng(v;rho;A.ds;A.da;de;e;x;...)
| . . | . . . | | . | | 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | . | | 29. ∀v:[[dec_lookup(A.da;kind(act))]] rho.
| . . | . . . | | . | | |   [[...]] rho A.ds ... de e x v ...
| . . | . . . | | . | | |   ⇒
| . . | . . . | | . | | |   [[wp(A;...;I)]] rho A.ds ... de e x v ...
| . . | . . . | | . | | 30. r : rel()
| . . | . . . | | . | | 31. r ∈ I ∨ r ∈ action_pre(kind(act);A.pre)
| . . | . . . | | . | | ⊢  [[r]] rho A.ds ... de e x value(act) ...
| . . | . . . | | . | |  by D (-1)
| . . | . . . | | . | |
| . . | . . . | | . | |\
| . . | . . . | | . | | 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. guarded_trace(A.da;te;I)
| . . | . . . | | . | | 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | . | | 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | . | | 20. [[A]] rho de e.init(s0)
| . . | . . . | | . | | 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | . | | 22. mk_trace_env(l, te)
| . . | . . . | | . | | |   ∈ trace_env([[A.da]] rho)
| . . | . . . | | . | | 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | | . | | 24. [[I]] rho A.ds <> de e x • ...
| . . | . . . | | . | | 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | | . | | 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | . | | 27. ∀v:vc{i:l}().
| . . | . . . | | . | | |   v ∈ <*vc_imp(mk_imp(A.init, I))*>
| . . | . . . | | . | | |   ∨ (∃a:dec().
| . . | . . . | | . | | |      a ∈ A.da & v = ioa_trans(A;a.lbl;I))
```

```
| . . | . . . | | . | | | |   ⇒
| . . | . . . | | . | | | |   vc_mng(v;rho;A.ds;A.da;de;e;x;...)
| . . | . . . | | . | | | 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | . | | | 29. ∀v:[[dec_lookup(A.da;kind(act))]] rho.
| . . | . . . | | . | | | |   [[...]] rho A.ds ... de e x v ...
| . . | . . . | | . | | | |   ⇒
| . . | . . . | | . | | | |   [[wp(A;...;I)]] rho ... ... de e x v ...
| . . | . . . | | . | | | 30. r : rel()
| . . | . . . | | . | | | 31. r ∈ I
| . . | . . . | | . | | | ⊢  [[r]] rho A.ds ... de e x value(act) ...
| . . | . . . | | . | | | | by AssertBY
| . . | . . . | | . | | | |    [[r]] rho A.ds <> de e x · ...
| . . | . . . | | . | | |
| . . | . . . | | . | | | |       (EasyHypThen Auto)
| . . | . . . | | . | | | |
| . . | . . . | | . | | | 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. guarded_trace(A.da;te;I)
| . . | . . . | | . | | | 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | . | | | 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | . | | | 20. [[A]] rho de e.init(s0)
| . . | . . . | | . | | | 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | . | | | 22. mk_trace_env(l, te)
| . . | . . . | | . | | | |   ∈ trace_env([[A.da]] rho)
| . . | . . . | | . | | | 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | | . | | | 24. [[I]] rho A.ds <> de e x · ...
| . . | . . . | | . | | | 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | | . | | | 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | . | | | 27. ∀v:vc{i:l}().
| . . | . . . | | . | | | |   v ∈ <*vc_imp(mk_imp(A.init, I))*>
| . . | . . . | | . | | | |   ∨ (∃a:dec().
| . . | . . . | | . | | | |      a ∈ A.da & v = ioa_trans(A;a.lbl;I))
| . . | . . . | | . | | | |   ⇒
| . . | . . . | | . | | | |   vc_mng(v;rho;A.ds;A.da;de;e;x;...)
| . . | . . . | | . | | | 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | . | | | 29. ∀v:[[dec_lookup(A.da;kind(act))]] rho.
| . . | . . . | | . | | | |   [[...]] rho A.ds ... de e x v ...
| . . | . . . | | . | | | |   ⇒
| . . | . . . | | . | | | |   [[wp(A;...;I)]] rho ... ... de e x v ...
| . . | . . . | | . | | | 30. r : rel()
| . . | . . . | | . | | | 31. r ∈ I
```

```
| . . | . . . | | . | | | 32. [[r]] rho A.ds <> de e x • ...
| . . | . . . | | . | | | ⊢  [[r]] rho A.ds ... de e x value(act) ...
| . . | . . . | | . | | | | by NthHypSq (-1)
| . . | . . . | | . | | | |
| . . | . . . | | . | | | .....assertion .....
| . . | . . . | | . | | | 1. A : ioa{i:l}()
| . . | . . . | | . | | | 2. I : Fmla
| . . | . . . | | . | | | 3. rho : Decl
| . . | . . . | | . | | | 4. de : sig()
| . . | . . . | | . | | | 5. e : {[[de]] rho}
| . . | . . . | | . | | | 6. te : Label→Label→ℙ
| . . | . . . | | . | | | 7. tc_ioa(A;de)
| . . | . . . | | . | | | 8. ¬ioa_mentions_trace(A)
| . . | . . . | | . | | | 9. trace_consistent_pred(rho;A.da;te;I)
| . . | . . . | | . | | | 10. guarded_trace(A.da;te;I)
| . . | . . . | | . | | | 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | . | | | 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | . | | | 20. [[A]] rho de e.init(s0)
| . . | . . . | | . | | | 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | . | | | 22. mk_trace_env(l, te)
| . . | . . . | | . | | |   ∈ trace_env([[A.da]] rho)
| . . | . . . | | . | | | 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | | . | | | 24. [[I]] rho A.ds <> de e x • ...
| . . | . . . | | . | | | 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | | . | | | 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | . | | | 27. ∀v:vc{i:l}().
| . . | . . . | | . | | |   v ∈ <*vc_imp(mk_imp(A.init, I))*>
| . . | . . . | | . | | |   ∨ (∃a:dec().
| . . | . . . | | . | | |      a ∈ A.da & v = ioa_trans(A;a.lbl;I))
| . . | . . . | | . | | |   ⇒
| . . | . . . | | . | | |   vc_mng(v;rho;A.ds;A.da;de;e;x;...)
| . . | . . . | | . | | | 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | . | | | 29. ∀v:[[dec_lookup(A.da;kind(act))]] rho.
| . . | . . . | | . | | |   [[...]] rho A.ds ... de e x v ...
| . . | . . . | | . | | |   ⇒
| . . | . . . | | . | | |   [[wp(A;...;I)]] rho ... ... de e x v ...
| . . | . . . | | . | | | 30. r : rel()
| . . | . . . | | . | | | 31. r ∈ I
| . . | . . . | | . | | | 32. [[r]] rho A.ds <> de e x • ...
| . . | . . . | | . | | | ⊢  [[r]] rho A.ds ... de e x ... ... ∼ ...
| . . | . . . | | . | | | | by ...
| . . | . . . | | . | | | |
| . . | . . . | | . | | | 1. A : ioa{i:l}()
| . . | . . . | | . | | | 2. I : Fmla
| . . | . . . | | . | | | 3. rho : Decl
| . . | . . . | | . | | | 4. de : sig()
| . . | . . . | | . | | | 5. e : {[[de]] rho}
| . . | . . . | | . | | | 6. te : Label→Label→ℙ
```

```
| . . | . . . | | . | | | 7. tc_ioa(A;de)
| . . | . . . | | . | | | 8. ¬ioa_mentions_trace(A)
| . . | . . . | | . | | | 9. trace_consistent_pred(rho;A.da;te;I)
| . . | . . . | | . | | | 10. guarded_trace(A.da;te;I)
| . . | . . . | | . | | | 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | . | | | 12. covers_pred(A;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. l : [[A]] rho de e.action List

| . . | . . . | | . | | | 20. [[A]] rho de e.init(s0)
| . . | . . . | | . | | | 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | . | | | 22. mk_trace_env(l, te)
| . . | . . . | | . | | |     ∈ trace_env([[A.da]] rho)
| . . | . . . | | . | | | 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | | . | | | 24. [[I]] rho A.ds <> de e x • ...
| . . | . . . | | . | | | 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | | . | | | 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | . | | | 27. ∀v:vc{i:l}().
| . . | . . . | | . | | |     v ∈ <*vc_imp(mk_imp(A.init, I))*>
| . . | . . . | | . | | |     ∨ (∃a:dec().
| . . | . . . | | . | | |        a ∈ A.da & v = ioa_trans(A;a.lbl;I))
| . . | . . . | | . | | |     ⇒
| . . | . . . | | . | | |     vc_mng(v;rho;A.ds;A.da;de;e;x;...)
| . . | . . . | | . | | | 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | . | | | 29. ∀v:[[dec_lookup(A.da;kind(act))]] rho.
| . . | . . . | | . | | |     [[...]] rho A.ds ... de e x v ...
| . . | . . . | | . | | |     ⇒
| . . | . . . | | . | | |     [[wp(A;...;I)]] rho ... ... de e x v ...
| . . | . . . | | . | | | 30. r : rel()
| . . | . . . | | . | | | 31. r ∈ I
| . . | . . . | | . | | | 32. [[r]] rho A.ds <> de e x • ...
| . . | . . . | | . | | | ⊢  closed_rel(r)  by EasyHypThen Auto
| . . | . . . | | . | | |
| . . | . . . | | . | .\
| . . | . . . | | . | . 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. guarded_trace(A.da;te;I)
| . . | . . . | | . | . 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | . | . 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | . | | . 20. [[A]] rho de e.init(s0)
| . . | . . . | | . | | . 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | . | | . 22. mk_trace_env(l, te)
| . . | . . . | | . | | . |   ∈ trace_env([[A.da]] rho)
| . . | . . . | | . | | . 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | | . | | . 24. [[I]] rho A.ds <> de e x • ...
| . . | . . . | | . | | . 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | | . | | . 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | . | | . 27. ∀v:vc{i:l}().
| . . | . . . | | . | | . |   v ∈ <*vc_imp(mk_imp(A.init, I))*>
| . . | . . . | | . | | . |   ∨ (∃a:dec().
| . . | . . . | | . | | . |      a ∈ A.da & v = ioa_trans(A;a.lbl;I))
| . . | . . . | | . | | . |   ⇒
| . . | . . . | | . | | . |   vc_mng(v;rho;A.ds;A.da;de;e;x;...)
| . . | . . . | | . | | . 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | . | | . 29. ∀v:[[dec_lookup(A.da;kind(act))]] rho.
| . . | . . . | | . | | . |   [[...]] rho A.ds ... de e x v ...
| . . | . . . | | . | | . |   ⇒
| . . | . . . | | . | | . |   [[wp(A;...;I)]] rho ... ... de e x v ...
| . . | . . . | | . | | . 30. r : rel()
| . . | . . . | | . | | . 31. r ∈ action_pre(kind(act);A.pre)
| . . | . . . | | . | | . ⊢  [[r]] rho A.ds ... de e x value(act) ...
| . . | . . . | | . | | . | by ...
| . . | . . . | | . | | . THEN EqLblFwd THEN SubstFor r 0
| . . | . . . | | . | | .
| . . | . . . | | . | | . |
| . . | . . . | | . | | . |
| . . | . . . | | . | | . |
| . . | . . . | | . | | . 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. guarded_trace(A.da;te;I)
| . . | . . . | | . | | . 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | . | | . 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | . | | . 20. [[A]] rho de e.init(s0)
| . . | . . . | | . | | . 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | . | | . 22. mk_trace_env(l, te)
| . . | . . . | | . | | . |   ∈ trace_env([[A.da]] rho)
| . . | . . . | | . | | . 23. t : dec()
| . . | . . . | | . | | . 24. t ∈ A.da
```

```
| . . | . . . | | . | | . 25. t.lbl = kind(act)
| . . | . . . | | . | | . 26. [[I]] rho A.ds <> de e x • ...
| . . | . . . | | . | | . 27. [[A]] rho de e.trans(x,act,x')

| . . | . . . | | . | | . 28. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | . | | . 29. ∀v:vc{i:l}().
| . . | . . . | | . | | . |   v ∈ <*vc_imp(mk_imp(A.init, I))*>
| . . | . . . | | . | | . |   ∨ (∃a:dec().
| . . | . . . | | . | | . |      a ∈ A.da & v = ioa_trans(A;a.lbl;I))
| . . | . . . | | . | | . |   ⇒
| . . | . . . | | . | | . |   vc_mng(v;rho;A.ds;A.da;de;e;x;...)
| . . | . . . | | . | | . 30. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | . | | . 31. ∀v:[[dec_lookup(A.da;kind(act))]] rho.
| . . | . . . | | . | | . |   [[...]] rho A.ds ... de e x v ...
| . . | . . . | | . | | . |   ⇒
| . . | . . . | | . | | . |   [[wp(A;...;I)]] rho ... ... de e x v ...
| . . | . . . | | . | | . 32. r : rel()
| . . | . . . | | . | | . 33. p : pre()
| . . | . . . | | . | | . 34. p ∈ A.pre
| . . | . . . | | . | | . 35. p.kind =1 kind(act)
| . . | . . . | | . | | . 36. r = p.rel
| . . | . . . | | . | | . 37. p.kind = kind(act)
| . . | . . . | | . | | . ⊢  [[p.rel]] rho A.ds ... de e x ... ...
| . . | . . . | | . | | . | by ...
| . . | . . . | | . | | . |                THEN InstHyp [p
| . . | . . . | | . | | . |    ] h THEN Auto
| . . | . . . | | . | |
| . . | . . . | | . | | . |    ...
| . . | . . . | | . | | . |
| . . | . . . | | . | | . 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. guarded_trace(A.da;te;I)
| . . | . . . | | . | | . 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | . | | . 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | . | | . 20. [[A]] rho de e.init(s0)
| . . | . . . | | . | | . 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | . | | . 22. mk_trace_env(l, te)
| . . | . . . | | . | | .     ∈ trace_env([[A.da]] rho)
| . . | . . . | | . | | . 23. t : dec()
| . . | . . . | | . | | . 24. t ∈ A.da
| . . | . . . | | . | | . 25. t.lbl = kind(act)
| . . | . . . | | . | | . 26. [[I]] rho A.ds <> de e x • ...
```

```
| . . | . . . | | . | | . 27. ∀p:pre().
| . . | . . . | | . | | .     p ∈ A.pre
| . . | . . . | | . | | .     ⇒
| . . | . . . | | . | | .     p.kind = kind(act)
| . . | . . . | | . | | .     ⇒
| . . | . . . | | . | | .     [[p.rel]] rho A.ds ... de e x ... ...
| . . | . . . | | . | | . 28. ∀ef:eff().
| . . | . . . | | . | | .     ef ∈ A.eff
| . . | . . . | | . | | .     ⇒
| . . | . . . | | . | | .     ef.kind = kind(act)
| . . | . . . | | . | | .     ⇒
| . . | . . . | | . | | .     x'.ef.smt.lbl
| . . | . . . | | . | | .     =
| . . | . . . | | . | | .     [[ef.smt.term]] 1of(e) x value(act) ...
| . . | . . . | | . | | . 29. ∀fr:frame().
| . . | . . . | | . | | .     fr ∈ A.frame
| . . | . . . | | . | | .     ⇒
| . . | . . . | | . | | .     ¬(kind(act) ∈ fr.acts)
| . . | . . . | | . | | .     ⇒
| . . | . . . | | . | | .     x'.fr.var = x.fr.var
| . . | . . . | | . | | . 30. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | . | | . 31. ∀v:vc{i:l}().
| . . | . . . | | . | | .     v ∈ <*vc_imp(mk_imp(A.init, I))*>
| . . | . . . | | . | | .     ∨ (∃a:dec().
| . . | . . . | | . | | .        a ∈ A.da & v = ioa_trans(A;a.lbl;I))
| . . | . . . | | . | | .     ⇒
| . . | . . . | | . | | .     vc_mng(v;rho;A.ds;A.da;de;e;x;...)
| . . | . . . | | . | | . 32. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | . | | . 33. ∀v:[[dec_lookup(A.da;kind(act))]] rho.
| . . | . . . | | . | | .     [[...]] rho A.ds ... de e x v ...
| . . | . . . | | . | | .     ⇒
| . . | . . . | | . | | .     [[wp(A;...;I)]] rho ... ... de e x v ...
| . . | . . . | | . | | . 34. r : rel()
| . . | . . . | | . | | . 35. p : pre()
| . . | . . . | | . | | . 36. p ∈ A.pre
| . . | . . . | | . | | . 37. p.kind =1 kind(act)
| . . | . . . | | . | | . 38. r = p.rel
| . . | . . . | | . | | . 39. p.kind = kind(act)
| . . | . . . | | . | | . 40. [[p.rel]] rho A.ds ... de e x ... ...
| . . | . . . | | . | | . ⊢  ¬rel_mentions_trace(p.rel)
| . . | . . . | | . | | .   by ...
| . . | . . . | | . | | . THEN OrLeft THENA Auto
| . . | . . . | | . | | .
| . . | . . . | | . | | .      THEN OrRight THENA Auto
| . . | . . . | | . | | .
| . . | . . . | | . | | .      THEN InstConcl [p] THEN Auto
| . . | . . . | | . | |
| . . | . . . | | . | |\

| . . | . . . | | . | | .....subterm .....
| . . | . . . | | . | | 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. guarded_trace(A.da;te;I)
| . . | . . . | | . | | 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | . | | 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | . | | 20. [[A]] rho de e.init(s0)
| . . | . . . | | . | | 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | . | | 22. mk_trace_env(l, te)
| . . | . . . | | . |     |     ∈ trace_env([[A.da]] rho)
| . . | . . . | | . | | 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | | . | | 24. [[I]] rho A.ds <> de e x · mk_trace_env(l,
| . . | . . . | | . |     |                                    te)
| . . | . . . | | . | | 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | | . | | 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | . | | 27. ∀v:vc{i:l}().
| . . | . . . | | . |     |     v ∈ <*vc_imp(mk_imp(A.init, I))*>
| . . | . . . | | . |     |     ∨ (∃a:dec().
| . . | . . . | | . |     |        a ∈ A.da & v = ioa_trans(A;a.lbl;I))
| . . | . . . | | . |     |     ⇒
| . . | . . . | | . |     |     vc_mng(v;rho;A.ds;A.da;de;e;x;...)
| . . | . . . | | . | | 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | . | | 29. ∀v:[[dec_lookup(A.da;kind(act))]] rho.
| . . | . . . | | . |     |     [[...]] rho A.ds ... de e x v ...
| . . | . . . | | . |     |     ⇒
| . . | . . . | | . |     |     [[wp(A;...;I)]] rho A.ds ... de e x v ...
| . . | . . . | | . | | 30. rel()
| . . | . . . | | . | | ⊢  I ∈ Collection(rel())
| . . | . . . | | . |     |   by Fold 'pred' 0 THEN Auto
| . . | . . . | | . |
| . . | . . . | | . | .\
| . . | . . . | | . | . .....subterm .....
| . . | . . . | | . | . 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. guarded_trace(A.da;te;I)
| . . | . . . | | . | . 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | . | . 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | . | . 20. [[A]] rho de e.init(s0)
| . . | . . . | | . | . 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | . | . 22. mk_trace_env(l, te)
| . . | . . . | | . | .     ∈ trace_env([[A.da]] rho)
| . . | . . . | | . | . 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | | . | . 24. [[I]] rho A.ds <> de e x · mk_trace_env(l,
| . . | . . . | | . | .                                te)
| . . | . . . | | . | . 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | | . | . 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | . | . 27. ∀v:vc{i:l}().
| . . | . . . | | . | .     v ∈ <*vc_imp(mk_imp(A.init, I))*>
| . . | . . . | | . | .     ∨ (∃a:dec().
| . . | . . . | | . | .        a ∈ A.da & v = ioa_trans(A;a.lbl;I))
| . . | . . . | | . | .     ⇒
| . . | . . . | | . | .     vc_mng(v;rho;A.ds;A.da;de;e;x;...)
| . . | . . . | | . | . 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | . | . 29. ∀v:[[dec_lookup(A.da;kind(act))]] rho.
| . . | . . . | | . | .     [[...]] rho A.ds ... de e x v ...
| . . | . . . | | . | .     ⇒
| . . | . . . | | . | .     [[wp(A;...;I)]] rho A.ds ... de e x v ...
| . . | . . . | | . | . 30. rel()
| . . | . . . | | . | . ⊢  action_pre(kind(act);A.pre)
| . . | . . . | | . | .    ∈ Collection(rel())
| . . | . . . | | . | .   by Fold 'pred' 0 THEN Auto
| . . | . . . | | . |
| . . | . . . | | . .\
| . . | . . . | | . . 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. guarded_trace(A.da;te;I)
| . . | . . . | | . . 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | . . 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | . . 20. [[A]] rho de e.init(s0)
| . . | . . . | | . . 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | . . 22. mk_trace_env(l, te)
| . . | . . . | | . . |   ∈ trace_env([[A.da]] rho)
| . . | . . . | | . . 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | | . . 24. [[I]] rho A.ds <> de e x · mk_trace_env(l,
| . . | . . . | | . . |                              te)
```

```
| . . | . . . | | . . 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | | . . 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | . . 27. ∀v:vc{i:l}().
| . . | . . . | | . . |   v ∈ <*vc_imp(mk_imp(A.init, I))*>
| . . | . . . | | . . |   ∨ (∃a:dec().
| . . | . . . | | . . |      a ∈ A.da & v = ioa_trans(A;a.lbl;I))
| . . | . . . | | . . |   ⇒
| . . | . . . | | . . |   vc_mng(v;rho;A.ds;A.da;de;e;x;...)
| . . | . . . | | . . 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | . . 29. ∀v:[[dec_lookup(A.da;kind(act))]] rho.
| . . | . . . | | . . |   [[...]] rho A.ds ... de e x v ...
| . . | . . . | | . . |   ⇒
| . . | . . . | | . . |   [[wp(A;...;I)]] rho A.ds ... de e x v ...
| . . | . . . | | . . 30. [[wp(A;...;I)]] rho A.ds ... de e x ... ...
| . . | . . . | | . . ⊢  [[wp(A;...;I)]] rho A.ds ... de e x ... ...
| . . | . . . | | . . | by Subst <kind(act),value(act)> ∼ act
| . . | . . . | | . . |     (-1) THEN Try Trivial
| . . | . . . | | . . |
| . . | . . . | | . . .....equality .....
| . . | . . . | | . . 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. guarded_trace(A.da;te;I)
| . . | . . . | | . . 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | | . . 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . | | . . 20. [[A]] rho de e.init(s0)
| . . | . . . | | . . 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | | . . 22. mk_trace_env(l, te)
| . . | . . . | | . .     ∈ trace_env([[A.da]] rho)
| . . | . . . | | . . 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | | . . 24. [[I]] rho A.ds <> de e x · mk_trace_env(l,
| . . | . . . | | . .                                 te)
| . . | . . . | | . . 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | | . . 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | | . . 27. ∀v:vc{i:l}().
| . . | . . . | | . .     v ∈ <*vc_imp(mk_imp(A.init, I))*>
| . . | . . . | | . .     ∨ (∃a:dec().
| . . | . . . | | . .        a ∈ A.da & v = ioa_trans(A;a.lbl;I))
| . . | . . . | | . .     ⇒
| . . | . . . | | . .     vc_mng(v;rho;A.ds;A.da;de;e;x;...)
| . . | . . . | | . . 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | | . . 29. ∀v:[[dec_lookup(A.da;kind(act))]] rho.
```

```
| . . | . . . | | . .     [[...]] rho A.ds ... de e x v ...
| . . | . . . | | . .     ⇒
| . . | . . . | | . .     [[wp(A;...;I)]] rho A.ds ... de e x v ...
| . . | . . . | | . . 30. [[wp(A;...;I)]] rho A.ds ... de e x ... ...
| . . | . . . | | . . ⊢  <kind(act),value(act)> ∼ act
| . . | . . . | | . .   by GenConcl act = z
| . . | . . . | | . .      ...
| . . | . . . | |
| . . | . . . | .\
| . . | . . . | . 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. guarded_trace(A.da;te;I)
| . . | . . . | . 11. tc_pred(I;A.ds;<>;de)
| . . | . . . | . 12. covers_pred(A;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. l : [[A]] rho de e.action List

| . . | . . . | . 20. [[A]] rho de e.init(s0)
| . . | . . . | . 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . | . 22. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | . . . | . 23. ∃t:dec(). t ∈ A.da & t.lbl = kind(act)
| . . | . . . | . 24. [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . | . . . | . 25. [[A]] rho de e.trans(x,act,x')
| . . | . . . | . 26. l ∈ (Σ[[A.da]] rho) List
| . . | . . . | . 27. [[VCs(A;I)]] rho A.ds A.da de e x ...
| . . | . . . | . 28. act ∈ (Σ[[A.da]] rho)
| . . | . . . | . 29. [[I]] rho A.ds <> de e x' · tappend(...;act)
| . . | . . . | .     ⇒
| . . | . . . | .     [[wp(A;kind(act);I)]] rho ... ... de e x ... ...
| . . | . . . | . 30. [[I]] rho A.ds <> de e x' · tappend(...;act)
| . . | . . . | . ⊢  [[I]] rho A.ds <> de e x' · mk_trace_env(l
| . . | . . . | .                                @ [act],
| . . | . . . | .                                te)
| . . | . . . | .   by ...
| . . | . . . |
| . . | . . . .\
| . . | . . . . 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. guarded_trace(A.da;te;I)
| . . | . . . . 11. tc_pred(I;A.ds;<>;de)
| . . | . . . . 12. covers_pred(A;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. l : [[A]] rho de e.action List
| . . | . . . . 20. [[A]] rho de e.init(s0)
| . . | . . . . 21. trace_reachable([[A]] rho de e;s0;l;x)
| . . | . . . . 22. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . | . . . . 23. [[I]] rho A.ds <> de e x · mk_trace_env(l, te)
| . . | . . . . 24. [[A]] rho de e.trans(x,act,x')
| . . | . . . . 25. l ∈ (Σ[[A.da]] rho) List
| . . | . . . . 26. [[VCs(A;I)]] rho A.ds A.da de e x mk_trace_env(l,
| . . | . . . .                                          te)
| . . | . . . . 27. act ∈ (Σ[[A.da]] rho)
| . . | . . . . 28. (∃t:dec(). t ∈ A.da & t.lbl = kind(act))
| . . | . . . .     ⇒
| . . | . . . .     [[I]] rho A.ds <> de e x' · mk_trace_env(l
| . . | . . . .                                  @ [act],
| . . | . . . .                                  te)
| . . | . . . . ⊢  [[I]] rho A.ds <> de e x' · mk_trace_env(l @ [act],
| . . | . . . .                                 te)
| . . | . . . .   by InstLemma `vc_trace_correct_action_decl_lemma`
| . . | . . . .
| . . | . . . .         [A;I;rho ;de ;e; te; s0;x;act; x';l
| . . | . . . .      ]  THENA (Auto THEN Reduce 0 THEN Auto)
| . . | . . . .
| . . | . . . .          THEN ThinTrivial
| . . | . . . .
| . . | . . . .      ...
| . . | . . . .
| . . |
| . . .\
| . . . 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. guarded_trace(A.da;te;I)
| . . . 11. tc_pred(I;A.ds;<>;de)
| . . . 12. covers_pred(A;I)
| . . . 13. closed_pred(I)
| . . . 14. single_valued_decls(A.ds)
| . . . 15. ∀s:[[A]] rho de e.state, tr:(Σ[[A.da]] rho) List.
| . . .     ([[A]] rho de e -tr-> s)
| . . .     ⇒
```

```
| . . .     [[VCs(A;I)]] rho A.ds A.da de e s mk_trace_env(tr, te)
| . . . 16. s0 : [[A]] rho de e.state
| . . . 17. x : [[A]] rho de e.state
| . . . 18. [[A]] rho de e.action
| . . . 19. [[A]] rho de e.state
| . . . 20. l : [[A]] rho de e.action List
| . . . 21. [[A]] rho de e.init(s0)
| . . . 22. trace_reachable([[A]] rho de e;s0;l;x)
| . . . 23. mk_trace_env(l, te) ∈ trace_env([[A.da]] rho)
| . . . ⊢  • ∈ [[<>]] rho  by BackThruLemma ‘empty_sts_mng‘ THEN Auto
|
.\
. 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. guarded_trace(A.da;te;I)
. 11. tc_pred(I;A.ds;<>;de)
. 12. covers_pred(A;I)
. 13. closed_pred(I)
. 14. single_valued_decls(A.ds)
. 15. s : [[A]] rho de e.state
. 16. tr : (Σ[[A.da]] rho) List
. 17. ([[A]] rho de e -tr-> s)
. ⊢  (∀v∈VCs(A;I).trace_consistent_vc(rho;A.da;mk_trace_env(tr,
.                                                te).proj;v))
.   by Reduce 0 THEN BackThruLemma ‘trace_consistent_ioa_inv_vc‘
.
.      THEN Auto
END
```