es-axioms THM

```
⊢  ∀the_es:ES.
|   (Trans e,e':E. (e <loc e'))
|   & SWellFounded((e <loc e'))
|   & (∀e,e':E. loc(e) = loc(e') ⟺ (e <loc e') ∨ e = e' ∨ (e' <loc e))
|   & (∀e:E. first(e) ⟺ (∀e':E. ¬(e' <loc e)))
|   & (∀e:E.
|      ¬first(e)
|      ⇒
|      (pred(e) <loc e) & (∀e':E. ¬((pred(e) <loc e') & (e' <loc e))))
|   & (∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e))))
|   & (Trans e,e':E. (e < e'))
|   & SWellFounded((e < e'))
|   & (∀e:E.
|      isrcv(e)
|      ⇒
|      sends(lnk(e);sender(e))[index(e)] = msg(lnk(e);tag(e);val(e)))
|   & (∀e,e':E. (e <loc e') ⇒ (e < e'))
|   & (∀e:E. isrcv(e) ⇒ (sender(e) < e))
|   & (∀e,e':E.
|      (e < e')
|      ⇒
|      ¬first(e') & (e < pred(e')) ∨ e = pred(e')
|      ∨ isrcv(e') & (e < sender(e')) ∨ e = sender(e'))
|   & (∀e:E. isrcv(e) ⇒ loc(e) = destination(lnk(e)))
|   & (∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil)
|   & (∀e,e':E.
|      isrcv(e)
|      ⇒
|      isrcv(e')
|      ⇒
|      lnk(e) = lnk(e')
|      ⇒
|      ((e <loc e')
|       ⟺
|       (sender(e) <loc sender(e'))
|       ∨ sender(e) = sender(e') & index(e)<index(e')))
|   & (∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
|      ∃e':E. isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n)
| by All_ES THEN Analyze -2 THEN ExRepD THEN All (Unfold `Msg_sub`)
|    THEN
|    SplitAndConcl
|    THEN
|    Try Trivial
|
|\
| 1. E : Type
| 2. EqDecider(E)
| 3. T : Id→Id→Type
| 4. V : Id→Id→Type
| 5. M : IdLnk→Id→Type
| 6. Top
| 7. loc : E→Id
| 8. kind : E→Knd
```

```
| 9. val : e:E→eventtype(kind;loc;V;M;e)
| 10. when : x:Id→e:E→T(loc(e),x)
| 11. after : x:Id→e:E→T(loc(e),x)
| 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
| 13. sender : {e:E| isrcv(kind(e)) }→E
| 14. index :
|     e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
| 15. first : E→𝔹
| 16. pred : {e':E| ¬first(e') }→E
| 17. causl : E→E→Prop
| 18. ∀e,e':E. loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
| 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
| 20. ∀e:E.
|     ¬first(e)
|     ⇒
|     loc(pred(e)) = loc(e) & causl(pred(e),e)
|     & (∀e':E.
|        loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
| 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
| 22. Trans e,e':E. causl(e,e')
| 23. SWellFounded(causl(e,e'))
| 24. ∀e:E.
|     isrcv(kind(e))
|     ⇒
|     (sends(lnk(kind(e)),sender(e)))[(index(e))]
|     =
|     msg(lnk(kind(e));tag(kind(e));val(e))
| 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
| 26. ∀e,e':E.
|     causl(e,e')
|     ⇒
|     ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
|     ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
| 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
| 29. ∀e,e':E.
|     isrcv(kind(e))
|     ⇒
|     isrcv(kind(e'))
|     ⇒
|     lnk(kind(e)) = lnk(kind(e'))
|     ⇒
|     (causl(e,e')
|      ⟺
|      causl(sender(e),sender(e'))
|      ∨ sender(e) = sender(e') & index(e)<index(e'))
| 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
|     ∃e':E.
|     isrcv(kind(e'))
|     & lnk(kind(e')) = l & sender(e') = e & index(e') = n
| 31. Top
| ⊢  Trans e,e':E. loc(e) = loc(e') & causl(e,e')
|   by AllHyps (λi.ParallelOp i THEN InstHyp [a;b;c] i)
|
|\
```

```
| 1. E : Type
| 2. EqDecider(E)
| 3. T : Id→Id→Type
| 4. V : Id→Id→Type
| 5. M : IdLnk→Id→Type
| 6. Top
| 7. loc : E→Id
| 8. kind : E→Knd
| 9. val : e:E→eventtype(kind;loc;V;M;e)
| 10. when : x:Id→e:E→T(loc(e),x)
| 11. after : x:Id→e:E→T(loc(e),x)
| 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
| 13. sender : {e:E| isrcv(kind(e)) }→E
| 14. index :
|     e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
| 15. first : E→𝔹
| 16. pred : {e':E| ¬first(e') }→E
| 17. causl : E→E→Prop
| 18. ∀e,e':E. loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
| 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
| 20. ∀e:E.
|     ¬first(e)
|     ⇒
|     loc(pred(e)) = loc(e) & causl(pred(e),e)
|     & (∀e':E.
|        loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
| 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
| 22. Trans e,e':E. causl(e,e')
| 23. SWellFounded(causl(e,e'))
| 24. ∀e:E.
|     isrcv(kind(e))
|     ⇒
|     (sends(lnk(kind(e)),sender(e)))[(index(e))]
|     =
|     msg(lnk(kind(e));tag(kind(e));val(e))
| 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
| 26. ∀e,e':E.
|     causl(e,e')
|     ⇒
|     ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
|     ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
| 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
| 29. ∀e,e':E.
|     isrcv(kind(e))
|     ⇒
|     isrcv(kind(e'))
|     ⇒
|     lnk(kind(e)) = lnk(kind(e'))
|     ⇒
|     (causl(e,e')
|      ⟺
|      causl(sender(e),sender(e'))
|      ∨ sender(e) = sender(e') & index(e)<index(e'))
| 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
```

```
|     ∃e':E.
|     isrcv(kind(e'))
|     & lnk(kind(e')) = l & sender(e') = e & index(e') = n
| 31. Top
| ⊢  SWellFounded(loc(e) = loc(e') & causl(e,e'))
|   by AllHyps
|      (λi.ParallelOp i THEN ParallelOp i THEN InstHyp [e;e'] i+1)
|
|\
| 1. E : Type
| 2. EqDecider(E)
| 3. T : Id→Id→Type
| 4. V : Id→Id→Type
| 5. M : IdLnk→Id→Type
| 6. Top
| 7. loc : E→Id
| 8. kind : E→Knd
| 9. val : e:E→eventtype(kind;loc;V;M;e)
| 10. when : x:Id→e:E→T(loc(e),x)
| 11. after : x:Id→e:E→T(loc(e),x)
| 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
| 13. sender : {e:E| isrcv(kind(e)) }→E
| 14. index :
|     e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
| 15. first : E→𝔹
| 16. pred : {e':E| ¬first(e') }→E
| 17. causl : E→E→Prop
| 18. ∀e,e':E. loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
| 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
| 20. ∀e:E.
|     ¬first(e)
|     ⇒
|     loc(pred(e)) = loc(e) & causl(pred(e),e)
|     & (∀e':E.
|        loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
| 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
| 22. Trans e,e':E. causl(e,e')
| 23. SWellFounded(causl(e,e'))
| 24. ∀e:E.
|     isrcv(kind(e))
|     ⇒
|     (sends(lnk(kind(e)),sender(e)))[(index(e))]
|     =
|     msg(lnk(kind(e));tag(kind(e));val(e))
| 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
| 26. ∀e,e':E.
|     causl(e,e')
|     ⇒
|     ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
|     ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
| 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
| 29. ∀e,e':E.
|     isrcv(kind(e))
|     ⇒
```

```
|     isrcv(kind(e'))
|     ⇒
|     lnk(kind(e)) = lnk(kind(e'))
|     ⇒
|     (causl(e,e')
|      ⟺
|      causl(sender(e),sender(e'))
|      ∨ sender(e) = sender(e') & index(e)<index(e'))
| 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
|     ∃e':E.
|     isrcv(kind(e'))
|     & lnk(kind(e')) = l & sender(e') = e & index(e') = n
| 31. Top
| ⊢  ∀e,e':E.
|    loc(e) = loc(e')
|    ⟺
|    loc(e) = loc(e') & causl(e,e')
|    ∨ e = e'
|    ∨ loc(e') = loc(e) & causl(e',e)
|   by AllHyps
|      (λi.
|       ParallelOp i THEN ParallelOp -1
|       THEN
|       Complete
|       (Auto
|        THENL
|        [ThinTrivial THEN Repeat (ParallelOp -1)
|        ;SplitOrHyps THEN HypSubst -1 0]))
|
|\
| 1. E : Type
| 2. EqDecider(E)
| 3. T : Id→Id→Type
| 4. V : Id→Id→Type
| 5. M : IdLnk→Id→Type
| 6. Top
| 7. loc : E→Id
| 8. kind : E→Knd
| 9. val : e:E→eventtype(kind;loc;V;M;e)
| 10. when : x:Id→e:E→T(loc(e),x)
| 11. after : x:Id→e:E→T(loc(e),x)
| 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
| 13. sender : {e:E| isrcv(kind(e)) }→E
| 14. index :
| |   e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
| 15. first : E→𝔹
| 16. pred : {e':E| ¬first(e') }→E
| 17. causl : E→E→Prop
| 18. ∀e,e':E. loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
| 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
| 20. ∀e:E.
| |   ¬first(e)
| |   ⇒
| |   loc(pred(e)) = loc(e) & causl(pred(e),e)
| |   & (∀e':E.
```

```
| |      loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
| 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
| 22. Trans e,e':E. causl(e,e')
| 23. SWellFounded(causl(e,e'))
| 24. ∀e:E.
| |   isrcv(kind(e))
| |   ⇒
| |   (sends(lnk(kind(e)),sender(e)))[(index(e))]
| |   =
| |   msg(lnk(kind(e));tag(kind(e));val(e))
| 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
| 26. ∀e,e':E.
| |   causl(e,e')
| |   ⇒
| |   ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
| |   ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
| 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
| 29. ∀e,e':E.
| |   isrcv(kind(e))
| |   ⇒
| |   isrcv(kind(e'))
| |   ⇒
| |   lnk(kind(e)) = lnk(kind(e'))
| |   ⇒
| |   (causl(e,e')
| |    ⟺
| |    causl(sender(e),sender(e'))
| |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
| |   ∃e':E.
| |   isrcv(kind(e'))
| |   & lnk(kind(e')) = l & sender(e') = e & index(e') = n
| 31. Top
| ⊢  ∀e:E. first(e) ⟺ (∀e':E. ¬(loc(e') = loc(e) & causl(e',e)))
| | by AllHyps
| |    (λi.
| |     ((ParallelOp i THEN RW (SweepDnC (HypC -1)) 0)
| |      THEN
| |      (Analyze 0)
| |      THEN
| |      (Analyze 0))
| |     THEN
| |     ParallelOp -1)
| |
| |\
| | 1. E : Type
| | 2. EqDecider(E)
| | 3. T : Id→Id→Type
| | 4. V : Id→Id→Type
| | 5. M : IdLnk→Id→Type
| | 6. Top
| | 7. loc : E→Id
| | 8. kind : E→Knd
| | 9. val : e:E→eventtype(kind;loc;V;M;e)
```

```
| | 10. when : x:Id→e:E→T(loc(e),x)
| | 11. after : x:Id→e:E→T(loc(e),x)
| | 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
| | 13. sender : {e:E| isrcv(kind(e)) }→E
| | 14. index :
| |     e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
| | 15. first : E→𝔹
| | 16. pred : {e':E| ¬first(e') }→E
| | 17. causl : E→E→Prop
| | 18. ∀e,e':E.
| |     loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
| | 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
| | 20. ∀e:E.
| |     ¬first(e)
| |     ⇒
| |     loc(pred(e)) = loc(e) & causl(pred(e),e)
| |     & (∀e':E.
| |        loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
| | 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
| | 22. Trans e,e':E. causl(e,e')
| | 23. SWellFounded(causl(e,e'))
| | 24. ∀e:E.
| |     isrcv(kind(e))
| |     ⇒
| |     (sends(lnk(kind(e)),sender(e)))[(index(e))]
| |     =
| |     msg(lnk(kind(e));tag(kind(e));val(e))
| | 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
| | 26. ∀e,e':E.
| |     causl(e,e')
| |     ⇒
| |     ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
| |     ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
| | 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
| | 29. ∀e,e':E.
| |     isrcv(kind(e))
| |     ⇒
| |     isrcv(kind(e'))
| |     ⇒
| |     lnk(kind(e)) = lnk(kind(e'))
| |     ⇒
| |     (causl(e,e')
| |      ⟺
| |      causl(sender(e),sender(e'))
| |      ∨ sender(e) = sender(e') & index(e)<index(e'))
| | 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
| |     ∃e':E.
| |     isrcv(kind(e'))
| |     & lnk(kind(e')) = l & sender(e') = e & index(e') = n
| | 31. Top
| | 32. e : E
| | 33. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
| | 34. ∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e)
| | 35. e' : E
```

```
| | 36. loc(e') = loc(e) ⇒ ¬causl(e',e)
| | ⊢  ¬(loc(e') = loc(e) & causl(e',e))
| |   by (Analyze 0) THEN (Analyze -2)
| |
| .\
| . 1. E : Type
| . 2. EqDecider(E)
| . 3. T : Id→Id→Type
| . 4. V : Id→Id→Type
| . 5. M : IdLnk→Id→Type
| . 6. Top
| . 7. loc : E→Id
| . 8. kind : E→Knd
| . 9. val : e:E→eventtype(kind;loc;V;M;e)
| . 10. when : x:Id→e:E→T(loc(e),x)
| . 11. after : x:Id→e:E→T(loc(e),x)
| . 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
| . 13. sender : {e:E| isrcv(kind(e)) }→E
| . 14. index :
| .     e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
| . 15. first : E→𝔹
| . 16. pred : {e':E| ¬first(e') }→E
| . 17. causl : E→E→Prop
| . 18. ∀e,e':E.
| .     loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
| . 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
| . 20. ∀e:E.
| .     ¬first(e)
| .     ⇒
| .     loc(pred(e)) = loc(e) & causl(pred(e),e)
| .     & (∀e':E.
| .        loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
| . 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
| . 22. Trans e,e':E. causl(e,e')
| . 23. SWellFounded(causl(e,e'))
| . 24. ∀e:E.
| .     isrcv(kind(e))
| .     ⇒
| .     (sends(lnk(kind(e)),sender(e)))[(index(e))]
| .     =
| .     msg(lnk(kind(e));tag(kind(e));val(e))
| . 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
| . 26. ∀e,e':E.
| .     causl(e,e')
| .     ⇒
| .     ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
| .     ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
| . 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
| . 29. ∀e,e':E.
| .     isrcv(kind(e))
| .     ⇒
| .     isrcv(kind(e'))
| .     ⇒
| .     lnk(kind(e)) = lnk(kind(e'))
```

```
| .     ⇒
| .     (causl(e,e')
| .      ⟺
| .      causl(sender(e),sender(e'))
| .      ∨ sender(e) = sender(e') & index(e)<index(e'))
| . 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
| .     ∃e':E.
| .     isrcv(kind(e'))
| .     & lnk(kind(e')) = l & sender(e') = e & index(e') = n
| . 31. Top
| . 32. e : E
| . 33. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
| . 34. ∀e':E. ¬(loc(e') = loc(e) & causl(e',e))
| . 35. e' : E
| . 36. ¬(loc(e') = loc(e) & causl(e',e))
| . ⊢  loc(e') = loc(e) ⇒ ¬causl(e',e)
| .   by (Auto THEN Analyze 0) THEN (Analyze -3)
|
|\
| 1. E : Type
| 2. EqDecider(E)
| 3. T : Id→Id→Type
| 4. V : Id→Id→Type
| 5. M : IdLnk→Id→Type
| 6. Top
| 7. loc : E→Id
| 8. kind : E→Knd
| 9. val : e:E→eventtype(kind;loc;V;M;e)
| 10. when : x:Id→e:E→T(loc(e),x)
| 11. after : x:Id→e:E→T(loc(e),x)
| 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
| 13. sender : {e:E| isrcv(kind(e)) }→E
| 14. index :
|     e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
| 15. first : E→𝔹
| 16. pred : {e':E| ¬first(e') }→E
| 17. causl : E→E→Prop
| 18. ∀e,e':E. loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
| 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
| 20. ∀e:E.
|     ¬first(e)
|     ⇒
|     loc(pred(e)) = loc(e) & causl(pred(e),e)
|     & (∀e':E.
|        loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
| 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
| 22. Trans e,e':E. causl(e,e')
| 23. SWellFounded(causl(e,e'))
| 24. ∀e:E.
|     isrcv(kind(e))
|     ⇒
|     (sends(lnk(kind(e)),sender(e)))[(index(e))]
|     =
|     msg(lnk(kind(e));tag(kind(e));val(e))
| 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
```

```
| 26. ∀e,e':E.
|      causl(e,e')
|      ⇒
|      ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
|      ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
| 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
| 29. ∀e,e':E.
|      isrcv(kind(e))
|      ⇒
|      isrcv(kind(e'))
|      ⇒
|      lnk(kind(e)) = lnk(kind(e'))
|      ⇒
|      (causl(e,e')
|       ⟺
|       causl(sender(e),sender(e'))
|       ∨ sender(e) = sender(e') & index(e)<index(e'))
| 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
|      ∃e':E.
|      isrcv(kind(e'))
|      & lnk(kind(e')) = l & sender(e') = e & index(e') = n
| 31. Top
| ⊢  ∀e:E.
|    ¬first(e)
|    ⇒
|    loc(pred(e)) = loc(e) & causl(pred(e),e)
|    & (∀e':E.
|       ¬(loc(pred(e)) = loc(e') & causl(pred(e),e')
|         & loc(e') = loc(e)
|         & causl(e',e)))
|   by AllHyps
|      (λi.
|       ((ParallelOp i THEN RepeatFor 3 (ParallelOp -1)
|         THEN
|         Analyze 0)
|        THEN
|        (Analyze -2)
|        THEN
|        (Analyze -1))
|       THEN
|       Complete Auto)
|
|\
| 1. E : Type
| 2. EqDecider(E)
| 3. T : Id→Id→Type
| 4. V : Id→Id→Type
| 5. M : IdLnk→Id→Type
| 6. Top
| 7. loc : E→Id
| 8. kind : E→Knd
| 9. val : e:E→eventtype(kind;loc;V;M;e)
| 10. when : x:Id→e:E→T(loc(e),x)
| 11. after : x:Id→e:E→T(loc(e),x)
```

```
| 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
| 13. sender : {e:E| isrcv(kind(e)) }→E
| 14. index :
|     e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
| 15. first : E→𝔹
| 16. pred : {e':E| ¬first(e') }→E
| 17. causl : E→E→Prop
| 18. ∀e,e':E. loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
| 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
| 20. ∀e:E.
|     ¬first(e)
|     ⇒
|     loc(pred(e)) = loc(e) & causl(pred(e),e)
|     & (∀e':E.
|        loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
| 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
| 22. Trans e,e':E. causl(e,e')
| 23. SWellFounded(causl(e,e'))
| 24. ∀e:E.
|     isrcv(kind(e))
|     ⇒
|     (sends(lnk(kind(e)),sender(e)))[(index(e))]
|     =
|     msg(lnk(kind(e));tag(kind(e));val(e))
| 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
| 26. ∀e,e':E.
|     causl(e,e')
|     ⇒
|     ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
|     ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
| 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
| 29. ∀e,e':E.
|     isrcv(kind(e))
|     ⇒
|     isrcv(kind(e'))
|     ⇒
|     lnk(kind(e)) = lnk(kind(e'))
|     ⇒
|     (causl(e,e')
|      ⟺
|      causl(sender(e),sender(e'))
|      ∨ sender(e) = sender(e') & index(e)<index(e'))
| 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
|     ∃e':E.
|     isrcv(kind(e'))
|     & lnk(kind(e')) = l & sender(e') = e & index(e') = n
| 31. Top
| ⊢  ∀e,e':E. loc(e) = loc(e') & causl(e,e') ⇒ causl(e,e')  by Auto
|
.\
. 1. E : Type
. 2. EqDecider(E)
. 3. T : Id→Id→Type
. 4. V : Id→Id→Type
```

```
. 5. M : IdLnk→Id→Type
. 6. Top
. 7. loc : E→Id
. 8. kind : E→Knd
. 9. val : e:E→eventtype(kind;loc;V;M;e)
. 10. when : x:Id→e:E→T(loc(e),x)
. 11. after : x:Id→e:E→T(loc(e),x)
. 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
. 13. sender : {e:E| isrcv(kind(e)) }→E
. 14. index :
. |   e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
. 15. first : E→𝔹
. 16. pred : {e':E| ¬first(e') }→E
. 17. causl : E→E→Prop
. 18. ∀e,e':E. loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
. 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
. 20. ∀e:E.
. |   ¬first(e)
. |   ⇒
. |   loc(pred(e)) = loc(e) & causl(pred(e),e)
. |   & (∀e':E.
. |      loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
. 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
. 22. Trans e,e':E. causl(e,e')
. 23. SWellFounded(causl(e,e'))
. 24. ∀e:E.
. |   isrcv(kind(e))
. |   ⇒
. |   (sends(lnk(kind(e)),sender(e)))[(index(e))]
. |   =
. |   msg(lnk(kind(e));tag(kind(e));val(e))
. 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
. 26. ∀e,e':E.
. |   causl(e,e')
. |   ⇒
. |   ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
. |   ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
. 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
. 29. ∀e,e':E.
. |   isrcv(kind(e))
. |   ⇒
. |   isrcv(kind(e'))
. |   ⇒
. |   lnk(kind(e)) = lnk(kind(e'))
. |   ⇒
. |   (causl(e,e')
. |    ⟺
. |    causl(sender(e),sender(e'))
. |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
. |   ∃e':E.
. |   isrcv(kind(e'))
. |   & lnk(kind(e')) = l & sender(e') = e & index(e') = n
. 31. Top
```

```
. ⊢  ∀e,e':E.
. |  isrcv(kind(e))
. |  ⇒
. |  isrcv(kind(e'))
. |  ⇒
. |  lnk(kind(e)) = lnk(kind(e'))
. |  ⇒
. |  (loc(e) = loc(e') & causl(e,e')
. |   ⟺
. |   loc(sender(e)) = loc(sender(e')) & causl(sender(e),sender(e'))
. |   ∨ sender(e) = sender(e') & index(e)<index(e'))
. | by ParallelOp 29 THEN RepeatFor 4 (ParallelOp -1)
. |
. |\
. | 1. E : Type
. | 2. EqDecider(E)
. | 3. T : Id→Id→Type
. | 4. V : Id→Id→Type
. | 5. M : IdLnk→Id→Type
. | 6. Top
. | 7. loc : E→Id
. | 8. kind : E→Knd
. | 9. val : e:E→eventtype(kind;loc;V;M;e)
. | 10. when : x:Id→e:E→T(loc(e),x)
. | 11. after : x:Id→e:E→T(loc(e),x)
. | 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
. | 13. sender : {e:E| isrcv(kind(e)) }→E
. | 14. index :
. | |   e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
. | 15. first : E→𝔹
. | 16. pred : {e':E| ¬first(e') }→E
. | 17. causl : E→E→Prop
. | 18. ∀e,e':E.
. | |   loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
. | 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
. | 20. ∀e:E.
. | |   ¬first(e)
. | |   ⇒
. | |   loc(pred(e)) = loc(e) & causl(pred(e),e)
. | |   & (∀e':E.
. | |      loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
. | 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
. | 22. Trans e,e':E. causl(e,e')
. | 23. SWellFounded(causl(e,e'))
. | 24. ∀e:E.
. | |   isrcv(kind(e))
. | |   ⇒
. | |   (sends(lnk(kind(e)),sender(e)))[(index(e))]
. | |   =
. | |   msg(lnk(kind(e));tag(kind(e));val(e))
. | 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
. | 26. ∀e,e':E.
. | |   causl(e,e')
. | |   ⇒
. | |   ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
```

```
. | |   ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
. | 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
. | 29. ∀e,e':E.
. | |   isrcv(kind(e))
. | |   ⇒
. | |   isrcv(kind(e'))
. | |   ⇒
. | |   lnk(kind(e)) = lnk(kind(e'))
. | |   ⇒
. | |   (causl(e,e')
. | |    ⟺
. | |    causl(sender(e),sender(e'))
. | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
. | |   ∃e':E.
. | |   isrcv(kind(e'))
. | |   & lnk(kind(e')) = l & sender(e') = e & index(e') = n
. | 31. Top
. | 32. e : E
. | 33. ∀e':E.
. | |   isrcv(kind(e))
. | |   ⇒
. | |   isrcv(kind(e'))
. | |   ⇒
. | |   lnk(kind(e)) = lnk(kind(e'))
. | |   ⇒
. | |   (causl(e,e')
. | |    ⟺
. | |    causl(sender(e),sender(e'))
. | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | 34. e' : E
. | 35. isrcv(kind(e))
. | 36. isrcv(kind(e'))
. | 37. lnk(kind(e)) = lnk(kind(e'))
. | 38. causl(e,e')
. | |   ⇒
. | |   causl(sender(e),sender(e'))
. | |   ∨ sender(e) = sender(e') & index(e)<index(e')
. | 39. (causl(e,e')) ⇐ (causl(sender(e),sender(e'))
. | |   ∨ sender(e) = sender(e') & index(e)<index(e'))
. | 40. loc(e) = loc(e')
. | 41. causl(e,e')
. | ⊢  loc(sender(e)) = loc(sender(e')) & causl(sender(e),sender(e'))
. | |  ∨ sender(e) = sender(e') & index(e)<index(e')
. | | by ThinTrivial THEN ParallelOp -1
. | |
. | 1. E : Type
. | 2. EqDecider(E)
. | 3. T : Id→Id→Type
. | 4. V : Id→Id→Type
. | 5. M : IdLnk→Id→Type
. | 6. Top
. | 7. loc : E→Id
. | 8. kind : E→Knd
```

```
. | 9. val : e:E→eventtype(kind;loc;V;M;e)
. | 10. when : x:Id→e:E→T(loc(e),x)
. | 11. after : x:Id→e:E→T(loc(e),x)
. | 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
. | 13. sender : {e:E| isrcv(kind(e)) }→E
. | 14. index :
. | |   e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
. | 15. first : E→𝔹
. | 16. pred : {e':E| ¬first(e') }→E
. | 17. causl : E→E→Prop
. | 18. ∀e,e':E.
. | |   loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
. | 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
. | 20. ∀e:E.
. | |   ¬first(e)
. | |   ⇒
. | |   loc(pred(e)) = loc(e) & causl(pred(e),e)
. | |   & (∀e':E.
. | |      loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
. | 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
. | 22. Trans e,e':E. causl(e,e')
. | 23. SWellFounded(causl(e,e'))
. | 24. ∀e:E.
. | |   isrcv(kind(e))
. | |   ⇒
. | |   (sends(lnk(kind(e)),sender(e)))[(index(e))]
. | |   =
. | |   msg(lnk(kind(e));tag(kind(e));val(e))
. | 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
. | 26. ∀e,e':E.
. | |   causl(e,e')
. | |   ⇒
. | |   ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
. | |   ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
. | 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
. | 29. ∀e,e':E.
. | |   isrcv(kind(e))
. | |   ⇒
. | |   isrcv(kind(e'))
. | |   ⇒
. | |   lnk(kind(e)) = lnk(kind(e'))
. | |   ⇒
. | |   (causl(e,e')
. | |    ⟺
. | |    causl(sender(e),sender(e'))
. | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
. | |   ∃e':E.
. | |   isrcv(kind(e'))
. | |   & lnk(kind(e')) = l & sender(e') = e & index(e') = n
. | 31. Top
. | 32. e : E
. | 33. ∀e':E.
. | |   isrcv(kind(e))
```

```
. | |   ⇒
. | |   isrcv(kind(e'))
. | |   ⇒
. | |   lnk(kind(e)) = lnk(kind(e'))
. | |   ⇒
. | |   (causl(e,e')
. | |    ⟺
. | |    causl(sender(e),sender(e'))
. | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | 34. e' : E
. | 35. isrcv(kind(e))
. | 36. isrcv(kind(e'))
. | 37. lnk(kind(e)) = lnk(kind(e'))
. | 38. (causl(e,e')) ⇐ (causl(sender(e),sender(e'))
. | |   ∨ sender(e) = sender(e') & index(e)<index(e'))
. | 39. loc(e) = loc(e')
. | 40. causl(e,e')
. | 41. causl(sender(e),sender(e'))
. | ⊢  loc(sender(e)) = loc(sender(e'))
. | | by Assert
. | |    (∀e:E.
. | |     isrcv(kind(e)) ⇒ loc(sender(e)) = source(lnk(kind(e))))
. | |
. | |\
. | | .....assertion .....
. | | 1. E : Type
. | | 2. EqDecider(E)
. | | 3. T : Id→Id→Type
. | | 4. V : Id→Id→Type
. | | 5. M : IdLnk→Id→Type
. | | 6. Top
. | | 7. loc : E→Id
. | | 8. kind : E→Knd
. | | 9. val : e:E→eventtype(kind;loc;V;M;e)
. | | 10. when : x:Id→e:E→T(loc(e),x)
. | | 11. after : x:Id→e:E→T(loc(e),x)
. | | 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
. | | 13. sender : {e:E| isrcv(kind(e)) }→E
. | | 14. index :
. | | |   e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
. | | 15. first : E→𝔹
. | | 16. pred : {e':E| ¬first(e') }→E
. | | 17. causl : E→E→Prop
. | | 18. ∀e,e':E.
. | | |   loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
. | | 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
. | | 20. ∀e:E.
. | | |   ¬first(e)
. | | |   ⇒
. | | |   loc(pred(e)) = loc(e) & causl(pred(e),e)
. | | |   & (∀e':E.
. | | |      loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
. | | 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
. | | 22. Trans e,e':E. causl(e,e')
. | | 23. SWellFounded(causl(e,e'))
```

```
. | | 24. ∀e:E.
. | | |   isrcv(kind(e))
. | | |   ⇒
. | | |   (sends(lnk(kind(e)),sender(e)))[(index(e))]
. | | |   =
. | | |   msg(lnk(kind(e));tag(kind(e));val(e))
. | | 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
. | | 26. ∀e,e':E.
. | | |   causl(e,e')
. | | |   ⇒
. | | |   ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
. | | |   ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
. | | 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | | 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
. | | 29. ∀e,e':E.
. | | |   isrcv(kind(e))
. | | |   ⇒
. | | |   isrcv(kind(e'))
. | | |   ⇒
. | | |   lnk(kind(e)) = lnk(kind(e'))
. | | |   ⇒
. | | |   (causl(e,e')
. | | |    ⟺
. | | |    causl(sender(e),sender(e'))
. | | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | | 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
. | | |   ∃e':E.
. | | |   isrcv(kind(e'))
. | | |   & lnk(kind(e')) = l & sender(e') = e & index(e') = n
. | | 31. Top
. | | 32. e : E
. | | 33. ∀e':E.
. | | |   isrcv(kind(e))
. | | |   ⇒
. | | |   isrcv(kind(e'))
. | | |   ⇒
. | | |   lnk(kind(e)) = lnk(kind(e'))
. | | |   ⇒
. | | |   (causl(e,e')
. | | |    ⟺
. | | |    causl(sender(e),sender(e'))
. | | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | | 34. e' : E
. | | 35. isrcv(kind(e))
. | | 36. isrcv(kind(e'))
. | | 37. lnk(kind(e)) = lnk(kind(e'))
. | | 38. (causl(e,e')) ⇐ (causl(sender(e),sender(e'))
. | | |   ∨ sender(e) = sender(e') & index(e)<index(e'))
. | | 39. loc(e) = loc(e')
. | | 40. causl(e,e')
. | | 41. causl(sender(e),sender(e'))
. | | ⊢  ∀e:E. isrcv(kind(e)) ⇒ loc(sender(e)) = source(lnk(kind(e)))
. | | | by Auto
. | | |
. | | 1. E : Type
```

```
. | | 2. EqDecider(E)
. | | 3. T : Id→Id→Type
. | | 4. V : Id→Id→Type
. | | 5. M : IdLnk→Id→Type
. | | 6. Top
. | | 7. loc : E→Id
. | | 8. kind : E→Knd
. | | 9. val : e:E→eventtype(kind;loc;V;M;e)
. | | 10. when : x:Id→e:E→T(loc(e),x)
. | | 11. after : x:Id→e:E→T(loc(e),x)
. | | 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
. | | 13. sender : {e:E| isrcv(kind(e)) }→E
. | | 14. index :
. | | |   e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
. | | 15. first : E→𝔹
. | | 16. pred : {e':E| ¬first(e') }→E
. | | 17. causl : E→E→Prop
. | | 18. ∀e,e':E.
. | | |   loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
. | | 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
. | | 20. ∀e:E.
. | | |   ¬first(e)
. | | |   ⇒
. | | |   loc(pred(e)) = loc(e) & causl(pred(e),e)
. | | |   & (∀e':E.
. | | |      loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
. | | 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
. | | 22. Trans e,e':E. causl(e,e')
. | | 23. SWellFounded(causl(e,e'))
. | | 24. ∀e:E.
. | | |   isrcv(kind(e))
. | | |   ⇒
. | | |   (sends(lnk(kind(e)),sender(e)))[(index(e))]
. | | |   =
. | | |   msg(lnk(kind(e));tag(kind(e));val(e))
. | | 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
. | | 26. ∀e,e':E.
. | | |   causl(e,e')
. | | |   ⇒
. | | |   ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
. | | |   ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
. | | 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | | 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
. | | 29. ∀e,e':E.
. | | |   isrcv(kind(e))
. | | |   ⇒
. | | |   isrcv(kind(e'))
. | | |   ⇒
. | | |   lnk(kind(e)) = lnk(kind(e'))
. | | |   ⇒
. | | |   (causl(e,e')
. | | |    ⟺
. | | |    causl(sender(e),sender(e'))
. | | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | | 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
```

```
. | | |   ∃e':E.
. | | |   isrcv(kind(e'))
. | | |   & lnk(kind(e')) = l & sender(e') = e & index(e') = n
. | | 31. Top
. | | 32. e : E
. | | 33. ∀e':E.
. | | |   isrcv(kind(e))
. | | |   ⇒
. | | |   isrcv(kind(e'))
. | | |   ⇒
. | | |   lnk(kind(e)) = lnk(kind(e'))
. | | |   ⇒
. | | |   (causl(e,e')
. | | |    ⟺
. | | |    causl(sender(e),sender(e'))
. | | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | | 34. e' : E
. | | 35. isrcv(kind(e))
. | | 36. isrcv(kind(e'))
. | | 37. lnk(kind(e)) = lnk(kind(e'))
. | | 38. (causl(e,e')) ⇐ (causl(sender(e),sender(e'))
. | | |   ∨ sender(e) = sender(e') & index(e)<index(e'))
. | | 39. loc(e) = loc(e')
. | | 40. causl(e,e')
. | | 41. causl(sender(e),sender(e'))
. | | 42. e1 : E
. | | 43. isrcv(kind(e1))
. | | ⊢  loc(sender(e1)) = source(lnk(kind(e1)))  by SupposeNot
. | | |
. | | 1. E : Type
. | | 2. EqDecider(E)
. | | 3. T : Id→Id→Type
. | | 4. V : Id→Id→Type
. | | 5. M : IdLnk→Id→Type
. | | 6. Top
. | | 7. loc : E→Id
. | | 8. kind : E→Knd
. | | 9. val : e:E→eventtype(kind;loc;V;M;e)
. | | 10. when : x:Id→e:E→T(loc(e),x)
. | | 11. after : x:Id→e:E→T(loc(e),x)
. | | 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
. | | 13. sender : {e:E| isrcv(kind(e)) }→E
. | | 14. index :
. | | |   e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
. | | 15. first : E→𝔹
. | | 16. pred : {e':E| ¬first(e') }→E
. | | 17. causl : E→E→Prop
. | | 18. ∀e,e':E.
. | | |   loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
. | | 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
. | | 20. ∀e:E.
. | | |   ¬first(e)
. | | |   ⇒
. | | |   loc(pred(e)) = loc(e) & causl(pred(e),e)
. | | |   & (∀e':E.
```

```
. | | |      loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
. | | 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
. | | 22. Trans e,e':E. causl(e,e')
. | | 23. SWellFounded(causl(e,e'))
. | | 24. ∀e:E.
. | | |   isrcv(kind(e))
. | | |   ⇒
. | | |   (sends(lnk(kind(e)),sender(e)))[(index(e))]
. | | |   =
. | | |   msg(lnk(kind(e));tag(kind(e));val(e))
. | | 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
. | | 26. ∀e,e':E.
. | | |   causl(e,e')
. | | |   ⇒
. | | |   ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
. | | |   ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
. | | 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | | 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
. | | 29. ∀e,e':E.
. | | |   isrcv(kind(e))
. | | |   ⇒
. | | |   isrcv(kind(e'))
. | | |   ⇒
. | | |   lnk(kind(e)) = lnk(kind(e'))
. | | |   ⇒
. | | |   (causl(e,e')
. | | |    ⟺
. | | |    causl(sender(e),sender(e'))
. | | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | | 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
. | | |   ∃e':E.
. | | |   isrcv(kind(e'))
. | | |   & lnk(kind(e')) = l & sender(e') = e & index(e') = n
. | | 31. Top
. | | 32. e : E
. | | 33. ∀e':E.
. | | |   isrcv(kind(e))
. | | |   ⇒
. | | |   isrcv(kind(e'))
. | | |   ⇒
. | | |   lnk(kind(e)) = lnk(kind(e'))
. | | |   ⇒
. | | |   (causl(e,e')
. | | |    ⟺
. | | |    causl(sender(e),sender(e'))
. | | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | | 34. e' : E
. | | 35. isrcv(kind(e))
. | | 36. isrcv(kind(e'))
. | | 37. lnk(kind(e)) = lnk(kind(e'))
. | | 38. (causl(e,e')) ⇐ (causl(sender(e),sender(e'))
. | | |   ∨ sender(e) = sender(e') & index(e)<index(e'))
. | | 39. loc(e) = loc(e')
. | | 40. causl(e,e')
. | | 41. causl(sender(e),sender(e'))
```

```
. | | 42. e1 : E
. | | 43. isrcv(kind(e1))
. | | 44. ¬loc(sender(e1)) = source(lnk(kind(e1)))
. | | ⊢  loc(sender(e1)) = source(lnk(kind(e1)))
. | | | by AssertBY (index(e1) ∈ ℕ||sends(lnk(kind(e1)),sender(e1))||)
. | | |    Auto
. | | |    THEN
. | | |    InstHyp [sender(e1);lnk(kind(e1))] 28
. | | |    THENA
. | | |    Complete Auto
. | | |
. | | 1. E : Type
. | | 2. EqDecider(E)
. | | 3. T : Id→Id→Type
. | | 4. V : Id→Id→Type
. | | 5. M : IdLnk→Id→Type
. | | 6. Top
. | | 7. loc : E→Id
. | | 8. kind : E→Knd
. | | 9. val : e:E→eventtype(kind;loc;V;M;e)
. | | 10. when : x:Id→e:E→T(loc(e),x)
. | | 11. after : x:Id→e:E→T(loc(e),x)
. | | 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
. | | 13. sender : {e:E| isrcv(kind(e)) }→E
. | | 14. index :
. | | |   e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
. | | 15. first : E→𝔹
. | | 16. pred : {e':E| ¬first(e') }→E
. | | 17. causl : E→E→Prop
. | | 18. ∀e,e':E.
. | | |   loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
. | | 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
. | | 20. ∀e:E.
. | | |   ¬first(e)
. | | |   ⇒
. | | |   loc(pred(e)) = loc(e) & causl(pred(e),e)
. | | |   & (∀e':E.
. | | |      loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
. | | 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
. | | 22. Trans e,e':E. causl(e,e')
. | | 23. SWellFounded(causl(e,e'))
. | | 24. ∀e:E.
. | | |   isrcv(kind(e))
. | | |   ⇒
. | | |   (sends(lnk(kind(e)),sender(e)))[(index(e))]
. | | |   =
. | | |   msg(lnk(kind(e));tag(kind(e));val(e))
. | | 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
. | | 26. ∀e,e':E.
. | | |   causl(e,e')
. | | |   ⇒
. | | |   ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
. | | |   ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
. | | 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | | 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
```

```
. | | 29. ∀e,e':E.
. | | |   isrcv(kind(e))
. | | |   ⇒
. | | |   isrcv(kind(e'))
. | | |   ⇒
. | | |   lnk(kind(e)) = lnk(kind(e'))
. | | |   ⇒
. | | |   (causl(e,e')
. | | |    ⟺
. | | |    causl(sender(e),sender(e'))
. | | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | | 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
. | | |   ∃e':E.
. | | |   isrcv(kind(e'))
. | | |   & lnk(kind(e')) = l & sender(e') = e & index(e') = n
. | | 31. Top
. | | 32. e : E
. | | 33. ∀e':E.
. | | |   isrcv(kind(e))
. | | |   ⇒
. | | |   isrcv(kind(e'))
. | | |   ⇒
. | | |   lnk(kind(e)) = lnk(kind(e'))
. | | |   ⇒
. | | |   (causl(e,e')
. | | |    ⟺
. | | |    causl(sender(e),sender(e'))
. | | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | | 34. e' : E
. | | 35. isrcv(kind(e))
. | | 36. isrcv(kind(e'))
. | | 37. lnk(kind(e)) = lnk(kind(e'))
. | | 38. (causl(e,e')) ⇐ (causl(sender(e),sender(e'))
. | | |   ∨ sender(e) = sender(e') & index(e)<index(e'))
. | | 39. loc(e) = loc(e')
. | | 40. causl(e,e')
. | | 41. causl(sender(e),sender(e'))
. | | 42. e1 : E
. | | 43. isrcv(kind(e1))
. | | 44. ¬loc(sender(e1)) = source(lnk(kind(e1)))
. | | 45. index(e1) ∈ ℕ||sends(lnk(kind(e1)),sender(e1))||
. | | 46. sends(lnk(kind(e1)),sender(e1)) = nil
. | | ⊢  loc(sender(e1)) = source(lnk(kind(e1)))
. | | | by ApFunToHypEquands 'Z' ||Z|| ℤ -1 THEN Reduce -1
. | | |
. | | 1. E : Type
. | | 2. EqDecider(E)
. | | 3. T : Id→Id→Type
. | | 4. V : Id→Id→Type
. | | 5. M : IdLnk→Id→Type
. | | 6. Top
. | | 7. loc : E→Id
. | | 8. kind : E→Knd
. | | 9. val : e:E→eventtype(kind;loc;V;M;e)
. | | 10. when : x:Id→e:E→T(loc(e),x)
```

```
. | | 11. after : x:Id→e:E→T(loc(e),x)
. | | 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
. | | 13. sender : {e:E| isrcv(kind(e)) }→E
. | | 14. index :
. | |     e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
. | | 15. first : E→𝔹
. | | 16. pred : {e':E| ¬first(e') }→E
. | | 17. causl : E→E→Prop
. | | 18. ∀e,e':E.
. | |     loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
. | | 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
. | | 20. ∀e:E.
. | |     ¬first(e)
. | |     ⇒
. | |     loc(pred(e)) = loc(e) & causl(pred(e),e)
. | |     & (∀e':E.
. | |        loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
. | | 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
. | | 22. Trans e,e':E. causl(e,e')
. | | 23. SWellFounded(causl(e,e'))
. | | 24. ∀e:E.
. | |     isrcv(kind(e))
. | |     ⇒
. | |     (sends(lnk(kind(e)),sender(e)))[(index(e))]
. | |     =
. | |     msg(lnk(kind(e));tag(kind(e));val(e))
. | | 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
. | | 26. ∀e,e':E.
. | |     causl(e,e')
. | |     ⇒
. | |     ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
. | |     ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
. | | 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | | 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
. | | 29. ∀e,e':E.
. | |     isrcv(kind(e))
. | |     ⇒
. | |     isrcv(kind(e'))
. | |     ⇒
. | |     lnk(kind(e)) = lnk(kind(e'))
. | |     ⇒
. | |     (causl(e,e')
. | |      ⟺
. | |      causl(sender(e),sender(e'))
. | |      ∨ sender(e) = sender(e') & index(e)<index(e'))
. | | 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
. | |     ∃e':E.
. | |     isrcv(kind(e'))
. | |     & lnk(kind(e')) = l & sender(e') = e & index(e') = n
. | | 31. Top
. | | 32. e : E
. | | 33. ∀e':E.
. | |     isrcv(kind(e))
. | |     ⇒
. | |     isrcv(kind(e'))
```

```
. | |     ⇒
. | |     lnk(kind(e)) = lnk(kind(e'))
. | |     ⇒
. | |     (causl(e,e')
. | |      ⟺
. | |      causl(sender(e),sender(e'))
. | |      ∨ sender(e) = sender(e') & index(e)<index(e'))
. | | 34. e' : E
. | | 35. isrcv(kind(e))
. | | 36. isrcv(kind(e'))
. | | 37. lnk(kind(e)) = lnk(kind(e'))
. | | 38. (causl(e,e')) ⇐ (causl(sender(e),sender(e'))
. | |     ∨ sender(e) = sender(e') & index(e)<index(e'))
. | | 39. loc(e) = loc(e')
. | | 40. causl(e,e')
. | | 41. causl(sender(e),sender(e'))
. | | 42. e1 : E
. | | 43. isrcv(kind(e1))
. | | 44. ¬loc(sender(e1)) = source(lnk(kind(e1)))
. | | 45. index(e1) ∈ ℕ||sends(lnk(kind(e1)),sender(e1))||
. | | 46. sends(lnk(kind(e1)),sender(e1)) = nil
. | | 47. ||sends(lnk(kind(e1)),sender(e1))|| = 0
. | | ⊢  loc(sender(e1)) = source(lnk(kind(e1)))
. | |   by HypSubst' -1 -3 THEN Reduce -3
. | |
. | .\
. | . 1. E : Type
. | . 2. EqDecider(E)
. | . 3. T : Id→Id→Type
. | . 4. V : Id→Id→Type
. | . 5. M : IdLnk→Id→Type
. | . 6. Top
. | . 7. loc : E→Id
. | . 8. kind : E→Knd
. | . 9. val : e:E→eventtype(kind;loc;V;M;e)
. | . 10. when : x:Id→e:E→T(loc(e),x)
. | . 11. after : x:Id→e:E→T(loc(e),x)
. | . 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
. | . 13. sender : {e:E| isrcv(kind(e)) }→E
. | . 14. index :
. | .     e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
. | . 15. first : E→𝔹
. | . 16. pred : {e':E| ¬first(e') }→E
. | . 17. causl : E→E→Prop
. | . 18. ∀e,e':E.
. | .     loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
. | . 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
. | . 20. ∀e:E.
. | .     ¬first(e)
. | .     ⇒
. | .     loc(pred(e)) = loc(e) & causl(pred(e),e)
. | .     & (∀e':E.
. | .        loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
. | . 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
. | . 22. Trans e,e':E. causl(e,e')
```

```
. | . 23. SWellFounded(causl(e,e'))
. | . 24. ∀e:E.
. | .     isrcv(kind(e))
. | .     ⇒
. | .     (sends(lnk(kind(e)),sender(e)))[(index(e))]
. | .     =
. | .     msg(lnk(kind(e));tag(kind(e));val(e))
. | . 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
. | . 26. ∀e,e':E.
. | .     causl(e,e')
. | .     ⇒
. | .     ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
. | .     ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
. | . 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
. | . 29. ∀e,e':E.
. | .     isrcv(kind(e))
. | .     ⇒
. | .     isrcv(kind(e'))
. | .     ⇒
. | .     lnk(kind(e)) = lnk(kind(e'))
. | .     ⇒
. | .     (causl(e,e')
. | .      ⟺
. | .      causl(sender(e),sender(e'))
. | .      ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
. | .     ∃e':E.
. | .     isrcv(kind(e'))
. | .     & lnk(kind(e')) = l & sender(e') = e & index(e') = n
. | . 31. Top
. | . 32. e : E
. | . 33. ∀e':E.
. | .     isrcv(kind(e))
. | .     ⇒
. | .     isrcv(kind(e'))
. | .     ⇒
. | .     lnk(kind(e)) = lnk(kind(e'))
. | .     ⇒
. | .     (causl(e,e')
. | .      ⟺
. | .      causl(sender(e),sender(e'))
. | .      ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . 34. e' : E
. | . 35. isrcv(kind(e))
. | . 36. isrcv(kind(e'))
. | . 37. lnk(kind(e)) = lnk(kind(e'))
. | . 38. (causl(e,e')) ⇐ (causl(sender(e),sender(e'))
. | .     ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . 39. loc(e) = loc(e')
. | . 40. causl(e,e')
. | . 41. causl(sender(e),sender(e'))
. | . 42. ∀e:E.
. | .     isrcv(kind(e)) ⇒ loc(sender(e)) = source(lnk(kind(e)))
. | . ⊢  loc(sender(e)) = loc(sender(e'))
```

```
. | .   by ((InstHyp [e] -1) THEN (InstHyp [e'] -2))
. | .      THEN
. | .      Subst (lnk(kind(e)) = lnk(kind(e'))) -2
. |
. |\
. | 1. E : Type
. | 2. EqDecider(E)
. | 3. T : Id→Id→Type
. | 4. V : Id→Id→Type
. | 5. M : IdLnk→Id→Type
. | 6. Top
. | 7. loc : E→Id
. | 8. kind : E→Knd
. | 9. val : e:E→eventtype(kind;loc;V;M;e)
. | 10. when : x:Id→e:E→T(loc(e),x)
. | 11. after : x:Id→e:E→T(loc(e),x)
. | 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
. | 13. sender : {e:E| isrcv(kind(e)) }→E
. | 14. index :
. |     e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
. | 15. first : E→𝔹
. | 16. pred : {e':E| ¬first(e') }→E
. | 17. causl : E→E→Prop
. | 18. ∀e,e':E.
. |     loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
. | 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
. | 20. ∀e:E.
. |     ¬first(e)
. |     ⇒
. |     loc(pred(e)) = loc(e) & causl(pred(e),e)
. |     & (∀e':E.
. |        loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
. | 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
. | 22. Trans e,e':E. causl(e,e')
. | 23. SWellFounded(causl(e,e'))
. | 24. ∀e:E.
. |     isrcv(kind(e))
. |     ⇒
. |     (sends(lnk(kind(e)),sender(e)))[(index(e))]
. |     =
. |     msg(lnk(kind(e));tag(kind(e));val(e))
. | 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
. | 26. ∀e,e':E.
. |     causl(e,e')
. |     ⇒
. |     ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
. |     ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
. | 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
. | 29. ∀e,e':E.
. |     isrcv(kind(e))
. |     ⇒
. |     isrcv(kind(e'))
. |     ⇒
. |     lnk(kind(e)) = lnk(kind(e'))
```

```
. |     ⇒
. |     (causl(e,e')
. |      ⟺
. |      causl(sender(e),sender(e'))
. |      ∨ sender(e) = sender(e') & index(e)<index(e'))
. | 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
. |     ∃e':E.
. |     isrcv(kind(e'))
. |     & lnk(kind(e')) = l & sender(e') = e & index(e') = n
. | 31. Top
. | 32. e : E
. | 33. ∀e':E.
. |     isrcv(kind(e))
. |     ⇒
. |     isrcv(kind(e'))
. |     ⇒
. |     lnk(kind(e)) = lnk(kind(e'))
. |     ⇒
. |     (causl(e,e')
. |      ⟺
. |      causl(sender(e),sender(e'))
. |      ∨ sender(e) = sender(e') & index(e)<index(e'))
. | 34. e' : E
. | 35. isrcv(kind(e))
. | 36. isrcv(kind(e'))
. | 37. lnk(kind(e)) = lnk(kind(e'))
. | 38. causl(e,e')
. |     ⇒
. |     causl(sender(e),sender(e'))
. |     ∨ sender(e) = sender(e') & index(e)<index(e')
. | 39. (causl(e,e')) ⇐ (causl(sender(e),sender(e'))
. |     ∨ sender(e) = sender(e') & index(e)<index(e'))
. | 40. loc(sender(e)) = loc(sender(e')) & causl(sender(e),sender(e'))
. |     ∨ sender(e) = sender(e') & index(e)<index(e')
. | ⊢  loc(e) = loc(e')
. |   by AllHyps
. |      (λi.
. |       InstHyp [e'] i THEN InstHyp [e] i
. |       THEN
. |       Subst (lnk(kind(e)) = lnk(kind(e'))) -1
. |       THEN
. |       Complete Auto)
. |
. .\
. . 1. E : Type
. . 2. EqDecider(E)
. . 3. T : Id→Id→Type
. . 4. V : Id→Id→Type
. . 5. M : IdLnk→Id→Type
. . 6. Top
. . 7. loc : E→Id
. . 8. kind : E→Knd
. . 9. val : e:E→eventtype(kind;loc;V;M;e)
. . 10. when : x:Id→e:E→T(loc(e),x)
. . 11. after : x:Id→e:E→T(loc(e),x)
```

```
. . 12. sends : l:IdLnk→E→({m:Msg(M)| haslink(l; m) } List)
. . 13. sender : {e:E| isrcv(kind(e)) }→E
. . 14. index :
. .     e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
. . 15. first : E→𝔹
. . 16. pred : {e':E| ¬first(e') }→E
. . 17. causl : E→E→Prop
. . 18. ∀e,e':E.
. .     loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
. . 19. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
. . 20. ∀e:E.
. .     ¬first(e)
. .     ⇒
. .     loc(pred(e)) = loc(e) & causl(pred(e),e)
. .     & (∀e':E.
. .        loc(e') = loc(e) ⇒ ¬(causl(pred(e),e') & causl(e',e)))
. . 21. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
. . 22. Trans e,e':E. causl(e,e')
. . 23. SWellFounded(causl(e,e'))
. . 24. ∀e:E.
. .     isrcv(kind(e))
. .     ⇒
. .     (sends(lnk(kind(e)),sender(e)))[(index(e))]
. .     =
. .     msg(lnk(kind(e));tag(kind(e));val(e))
. . 25. ∀e:E. isrcv(kind(e)) ⇒ causl(sender(e),e)
. . 26. ∀e,e':E.
. .     causl(e,e')
. .     ⇒
. .     ¬first(e') & causl(e,pred(e')) ∨ e = pred(e')
. .     ∨ isrcv(kind(e')) & causl(e,sender(e')) ∨ e = sender(e')
. . 27. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . 28. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l,e) = nil
. . 29. ∀e,e':E.
. .     isrcv(kind(e))
. .     ⇒
. .     isrcv(kind(e'))
. .     ⇒
. .     lnk(kind(e)) = lnk(kind(e'))
. .     ⇒
. .     (causl(e,e')
. .      ⟺
. .      causl(sender(e),sender(e'))
. .      ∨ sender(e) = sender(e') & index(e)<index(e'))
. . 30. ∀e:E, l:IdLnk, n:ℕ||sends(l,e)||.
. .     ∃e':E.
. .     isrcv(kind(e'))
. .     & lnk(kind(e')) = l & sender(e') = e & index(e') = n
. . 31. Top
. . 32. e : E
. . 33. ∀e':E.
. .     isrcv(kind(e))
. .     ⇒
. .     isrcv(kind(e'))
. .     ⇒
```

```
. .     lnk(kind(e)) = lnk(kind(e'))
. .     ⇒
. .     (causl(e,e')
. .      ⟺
. .      causl(sender(e),sender(e'))
. .      ∨ sender(e) = sender(e') & index(e)<index(e'))
. . 34. e' : E
. . 35. isrcv(kind(e))
. . 36. isrcv(kind(e'))
. . 37. lnk(kind(e)) = lnk(kind(e'))
. . 38. causl(e,e')
. .     ⇒
. .     causl(sender(e),sender(e'))
. .     ∨ sender(e) = sender(e') & index(e)<index(e')
. . 39. (causl(e,e')) ⇐ (causl(sender(e),sender(e'))
. .     ∨ sender(e) = sender(e') & index(e)<index(e'))
. . 40. loc(sender(e)) = loc(sender(e')) & causl(sender(e),sender(e'))
. .     ∨ sender(e) = sender(e') & index(e)<index(e')
. . ⊢  causl(e,e')  by Analyze -2 THEN ParallelOp -1
END
```