```
ESAxioms_wf  THM

⊢  ∀E:Type, T,V:(Id→Id→Type), M:(IdLnk→Id→Type), loc:(E→Id),
|   kind:(E→Knd), val:(e:E→eventtype(kind;loc;V;M;e)),
|   when,after:(x:Id→e:E→T(loc(e),x)),
|   sends:(l:IdLnk→E→(Msg_sub(l; M) List)), sender:({e:E
|                                                    | isrcv(kind
|                                                           (e)) }
|                                                   →E),
|   index:(e:{e:E| isrcv(kind(e)) }→ℕ||sends
|                                    (lnk(kind(e))
|                                    ,sender(e))||),
|   first:(E→𝔹), pred:({e':E| ¬first(e') }→E), causl:(E→E→Prop).
|  ESAxioms{i:l}
|          (E;
|           T;
|           M;
|           loc;
|           kind;
|           val;
|           when;
|           after;
|           sends;
|           sender;
|           index;
|           first;
|           pred;
|           causl)
|  ∈ Prop
| by Auto THEN Try (Analyze -1 THEN Unhide) THEN Unfold `ESAxioms` 0
|    THEN
|    Unfold `and` 0
|    THEN
|    Fold `cand` 0
|    THEN
|    Try (Unfold `Msg_sub` -1 THEN Analyze -1 THEN Complete Auto)
|
|\
| .....subterm .....
| 1. E : Type
| 2. T : Id→Id→Type
| 3. V : Id→Id→Type
| 4. M : IdLnk→Id→Type
| 5. loc : E→Id
| 6. kind : E→Knd
| 7. e:E→eventtype(kind;loc;V;M;e)
| 8. x:Id→e:E→T(loc(e),x)
| 9. after : x:Id→e:E→T(loc(e),x)
| 10. sends : l:IdLnk→E→(Msg_sub(l; M) List)
| 11. sender : {e:E| isrcv(kind(e)) }→E
| 12. e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
| 13. first : E→𝔹
| 14. pred : {e':E| ¬first(e') }→E
| 15. causl : E→E→Prop
| 16. ∀e,e':E. loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
```

```
| 17. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
| 18. ∀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)))
| 19. e : E
| 20. ¬first(e)
| 21. x : Id
| ⊢  after(x,pred(e)) ∈ T(loc(e),x)
| | by Assert (T(loc(pred(e)),x) = T(loc(e),x))
| |    THEN
| |    Assert (loc(e) = loc(pred(e)))
| |
| |\
| | .....assertion .....
| | 1. E : Type
| | 2. T : Id→Id→Type
| | 3. V : Id→Id→Type
| | 4. M : IdLnk→Id→Type
| | 5. loc : E→Id
| | 6. kind : E→Knd
| | 7. e:E→eventtype(kind;loc;V;M;e)
| | 8. x:Id→e:E→T(loc(e),x)
| | 9. x:Id→e:E→T(loc(e),x)
| | 10. sends : l:IdLnk→E→(Msg_sub(l; M) List)
| | 11. sender : {e:E| isrcv(kind(e)) }→E
| | 12. e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
| | 13. first : E→𝔹
| | 14. pred : {e':E| ¬first(e') }→E
| | 15. causl : E→E→Prop
| | 16. ∀e,e':E.
| |     loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
| | 17. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
| | 18. ∀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)))
| | 19. e : E
| | 20. ¬first(e)
| | 21. Id
| | ⊢  loc(e) = loc(pred(e))  by InstHyp [e] -4 THEN ExRepD
| |
| .\
| . 1. E : Type
| . 2. T : Id→Id→Type
| . 3. V : Id→Id→Type
| . 4. M : IdLnk→Id→Type
| . 5. loc : E→Id
| . 6. kind : E→Knd
| . 7. e:E→eventtype(kind;loc;V;M;e)
| . 8. x:Id→e:E→T(loc(e),x)
```

```
| . 9. x:Id→e:E→T(loc(e),x)
| . 10. sends : l:IdLnk→E→(Msg_sub(l; M) List)
| . 11. sender : {e:E| isrcv(kind(e)) }→E
| . 12. e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
| . 13. first : E→𝔹
| . 14. pred : {e':E| ¬first(e') }→E
| . 15. causl : E→E→Prop
| . 16. ∀e,e':E.
| .     loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
| . 17. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
| . 18. ∀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)))
| . 19. e : E
| . 20. ¬first(e)
| . 21. x : Id
| . 22. loc(e) = loc(pred(e))
| . ⊢  T(loc(pred(e)),x) = T(loc(e),x)  by HypSubst -1 0
|
.\
. .....subterm .....
. 1. E : Type
. 2. T : Id→Id→Type
. 3. V : Id→Id→Type
. 4. M : IdLnk→Id→Type
. 5. loc : E→Id
. 6. kind : E→Knd
. 7. val : e:E→eventtype(kind;loc;V;M;e)
. 8. when : x:Id→e:E→T(loc(e),x)
. 9. after : x:Id→e:E→T(loc(e),x)
. 10. sends : l:IdLnk→E→(Msg_sub(l; M) List)
. 11. sender : {e:E| isrcv(kind(e)) }→E
. 12. e:{e:E| isrcv(kind(e)) }→ℕ||sends(lnk(kind(e)),sender(e))||
. 13. first : E→𝔹
. 14. pred : {e':E| ¬first(e') }→E
. 15. causl : E→E→Prop
. 16. ∀e,e':E. loc(e) = loc(e') ⇒ causl(e,e') ∨ e = e' ∨ causl(e',e)
. 17. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬causl(e',e))
. 18. ∀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)))
. 19. ∀e:E. ¬first(e) ⇒ (∀x:Id. when(x,e) = after(x,pred(e)))
. 20. Trans e,e':E. causl(e,e')
. 21. SWellFounded(causl(e,e'))
. 22. e : E
. 23. isrcv(kind(e))
. ⊢  val(e) ∈ M(lnk(kind(e)),tag(kind(e)))
.   by DoSubsume THEN MoveToConcl -1 THEN Unfold `eventtype` 0
.      THEN
```

```
.       ConclKindCase (kind(e))
.       THEN
.       Analyze 0
END
```