```
finite-support-feasible  THM

⊢  ∀D:Dsys, L:Id List.
|  (∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i)))
|  ⇒
|  (∀i:Id. Feasible(M(i)))
|  ⇒
|  (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
|                                   ⇒
|  interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg)))))
|  ⇒
|  d-feasible(D)
| by Auto THEN Analyze 0 THEN Try Trivial THEN Analyze 0
|
|\
| 1. D : Dsys
| 2. L : Id List
| 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| 4. ∀i:Id. Feasible(M(i))
| 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
| |                                 ⇒
| |  interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg)))))
| ⊢  ∀l:IdLnk, tg:Id.
| |  M(source(l)).dout(l,tg) ⊆r M(destination(l)).din(l,tg)
| | by Auto THEN Decide deq-member(IdDeq;source(l);L)
| |    THEN
| |    RWO
| |    Thm* ∀eq:EqDecider(A), L:A List, x:A.
| |         deq-member(eq;x;L) ⟺ (x ∈ L)
| |    {assert_DASH_deq_DASH_member}
| |    -1
| |
| |\
| | 1. D : Dsys
| | 2. L : Id List
| | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | 4. ∀i:Id. Feasible(M(i))
| | 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
| | |                                 ⇒
| | |  interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
| | 6. l : IdLnk
| | 7. tg : Id
| | 8. (source(l) ∈ L)
| | ⊢  M(source(l)).dout(l,tg) ⊆r M(destination(l)).din(l,tg)
| | | by Decide deq-member(IdDeq;destination(l);L)
| | |    THEN
| | |    RWO
| | |    Thm* ∀eq:EqDecider(A), L:A List, x:A.
| | |         deq-member(eq;x;L) ⟺ (x ∈ L)
| | |    {assert_DASH_deq_DASH_member}
| | |    -1
| | |
| | |\
| | | 1. D : Dsys
| | | 2. L : Id List
```

```
| | | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | 4. ∀i:Id. Feasible(M(i))
| | | 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
| | | |                                    ⇒
| | | |  interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
| | | 6. l : IdLnk
| | | 7. tg : Id
| | | 8. (source(l) ∈ L)
| | | 9. (destination(l) ∈ L)
| | | ⊢  M(source(l)).dout(l,tg) ⊆r M(destination(l)).din(l,tg)
| | | | by Unfold ‘ma-dout‘ 0 THEN Unfold ‘fpf-cap‘ 0
| | | |    THEN
| | | |    SplitOnConclITE
| | | |
| | | |\
| | | | .....subterm .....
| | | | 1. D : Dsys
| | | | 2. L : Id List
| | | | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | | 4. ∀i:Id. Feasible(M(i))
| | | | 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
| | | |                                    ⇒
| | | |    interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
| | | | 6. l : IdLnk
| | | | 7. Id
| | | | 8. (source(l) ∈ L)
| | | | 9. (destination(l) ∈ L)
| | | | ⊢  1of(2of(M(source(l)))) ∈ a:Knd fp-> Top
| | | |   by GenConclMsgA THEN Unfold_MsgA -2
| | | |
| | | |\
| | | | .....truecase .....
| | | | 1. D : Dsys
| | | | 2. L : Id List
| | | | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | | 4. ∀i:Id. Feasible(M(i))
| | | | 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
| | | | |                                    ⇒
| | | | |  interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
| | | | 6. l : IdLnk
| | | | 7. tg : Id
| | | | 8. (source(l) ∈ L)
| | | | 9. (destination(l) ∈ L)
| | | | 10. rcv(l; tg) ∈ dom(1of(2of(M(source(l)))))
| | | | ⊢  1of(2of(M(source(l))))(rcv
| | | | |  (l; tg)) ⊆r M(destination(l)).din(l,tg)
| | | | | by Unfold ‘l_all‘ 5
| | | | |    THEN
| | | | |    InstHyp
| | | | |    [source(l);<l,tg,1of(2of(M(source(l))))(rcv(l; tg))>]
| | | | |    5
| | | | |    THEN
| | | | |    All Reduce
| | | | |
| | | | |\
```

```
| | | | | .....subterm .....
| | | | | 1. D : Dsys
| | | | | 2. L : Id List
| | | | | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | | | 4. ∀i:Id. Feasible(M(i))
| | | | | 5. ∀i:Id.
| | | | |    (i ∈ L)
| | | | |    ⇒
| | | | |    (∀ltg:(IdLnk×Id×Type).
| | | | |     (ltg ∈ ma-outlinks(M(i);i))
| | | | |     ⇒
| | | | |     (destination(1of(ltg)) ∈ L)
| | | | |     ⇒
| | | | |     interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
| | | | | 6. l : IdLnk
| | | | | 7. tg : Id
| | | | | 8. (source(l) ∈ L)
| | | | | 9. (destination(l) ∈ L)
| | | | | 10. rcv(l; tg) ∈ dom(1of(2of(M(source(l)))))
| | | | | ⊢  1of(2of(M(source(l))))(rcv(l; tg)) ∈ Type
| | | | |   by MoveToConcl -1 THEN GenConclMsgA THEN Unfold_MsgA -2
| | | | |
| | | | |\
| | | | | 1. D : Dsys
| | | | | 2. L : Id List
| | | | | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | | | 4. ∀i:Id. Feasible(M(i))
| | | | | 5. ∀i:Id.
| | | | | |  (i ∈ L)
| | | | | |  ⇒
| | | | | |  (∀ltg:(IdLnk×Id×Type).
| | | | | |   (ltg ∈ ma-outlinks(M(i);i))
| | | | | |   ⇒
| | | | | |   (destination(1of(ltg)) ∈ L)
| | | | | |   ⇒
| | | | | |   interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
| | | | | 6. l : IdLnk
| | | | | 7. tg : Id
| | | | | 8. (source(l) ∈ L)
| | | | | 9. (destination(l) ∈ L)
| | | | | 10. rcv(l; tg) ∈ dom(1of(2of(M(source(l)))))
| | | | | ⊢  (<l
| | | | | |   ,tg
| | | | | |   ,1of(2of(M(source(l))))(rcv
| | | | | |   (l; tg))> ∈ ma-outlinks(M(source(l));source(l)))
| | | | | | by MoveToConcl -1 THEN Unfold ‘ma-outlinks‘ 0
| | | | | |    THEN
| | | | | |    GenConclMsgA
| | | | | |    THEN
| | | | | |    Unfold_MsgA -2
| | | | | |
| | | | | 1. D : Dsys
| | | | | 2. L : Id List
| | | | | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | | | 4. ∀i:Id. Feasible(M(i))
```

```
| | | | | 5. ∀i:Id.
| | | | | |  (i ∈ L)
| | | | | |  ⇒
| | | | | |  (∀ltg:(IdLnk×Id×Type).
| | | | | |   (ltg ∈ ma-outlinks(M(i);i))
| | | | | |   ⇒
| | | | | |   (destination(1of(ltg)) ∈ L)
| | | | | |   ⇒
| | | | | |   interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
| | | | | 6. l : IdLnk
| | | | | 7. tg : Id
| | | | | 8. (source(l) ∈ L)
| | | | | 9. (destination(l) ∈ L)
| | | | | 10. ds : x:Id fp-> Type
| | | | | 11. da : a:Knd fp-> Type
| | | | | 12. init : x:Id fp-> ds(x)?Void
| | | | | 13. pre : a:Id fp-> State(ds)→da(locl(a))?Top→Prop
| | | | | 14. ef :
| | | | | |   kx:Knd×Id fp-> State(ds)→da(1of(kx))?Top→
| | | | | |   ds(2of(kx))?Void
| | | | | 15. send :
| | | | | |   kl:Knd×IdLnk fp-> (tg:Id
| | | | | |                      ×State(ds)→da(1of(kl))?Top→
| | | | | |                      (da(rcv(2of(kl); tg
| | | | | |                              ))?Void List)) List
| | | | | 16. frame : x:Id fp-> Knd List
| | | | | 17. sframe : ltg:IdLnk×Id fp-> Knd List
| | | | | 18. M8 : Top
| | | | | 19. M(source(l)) = <ds,da,init,pre,ef,send,frame,sframe,M8>
| | | | | 20. rcv(l; tg) ∈ dom(da)
| | | | | ⊢  (<l,tg,da(rcv(l; tg))> ∈ da-outlinks(da;source(l)))
| | | | | | by Unfold ‘da-outlinks‘ 0
| | | | | |
| | | | | 1. D : Dsys
| | | | | 2. L : Id List
| | | | | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | | | 4. ∀i:Id. Feasible(M(i))

| | | | | 5. ∀i:Id.
| | | | | |  (i ∈ L)
| | | | | |  ⇒
| | | | | |  (∀ltg:(IdLnk×Id×Type).
| | | | | |   (ltg ∈ ma-outlinks(M(i);i))
| | | | | |   ⇒
| | | | | |   (destination(1of(ltg)) ∈ L)
| | | | | |   ⇒
| | | | | |   interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
| | | | | 6. l : IdLnk
| | | | | 7. tg : Id
| | | | | 8. (source(l) ∈ L)
| | | | | 9. (destination(l) ∈ L)
| | | | | 10. ds : x:Id fp-> Type
| | | | | 11. da : a:Knd fp-> Type
| | | | | 12. init : x:Id fp-> ds(x)?Void
| | | | | 13. pre : a:Id fp-> State(ds)→da(locl(a))?Top→Prop
| | | | | 14. ef :
```

```
| | | | | | |   kx:Knd×Id fp-> State(ds)→da(1of(kx))?Top→
| | | | | | |   ds(2of(kx))?Void
| | | | | | 15. send :
| | | | | | |   kl:Knd×IdLnk fp-> (tg:Id
| | | | | | |                      ×State(ds)→da(1of(kl))?Top→
| | | | | | |                      (da(rcv(2of(kl); tg
| | | | | | |                              ))?Void List)) List
| | | | | | 16. frame : x:Id fp-> Knd List
| | | | | | 17. sframe : ltg:IdLnk×Id fp-> Knd List
| | | | | | 18. M8 : Top
| | | | | | 19. M(source(l)) = <ds,da,init,pre,ef,send,frame,sframe,M8>
| | | | | | 20. rcv(l; tg) ∈ dom(da)
| | | | | | ⊢  (<l,tg,da(rcv(l; tg))> ∈ mapfilter(λk.
| | | | | | |  da-outlink-f(da;k);λk.
| | | | | | |  has-src(source(l);k);fpf-dom-list(da)))
| | | | | | by Inst
| | | | | | |    Thm* ∀P:(T→𝔹), T':Type, f:({x:T| P(x) }→T'),
| | | | | | |          L:T List, x:T'.
| | | | | | |         (x ∈ mapfilter(f;P;L))
| | | | | | |         ⟺
| | | | | | |         (∃y:T. (y ∈ L) & P(y) & x = f(y))
| | | | | | |    {member_map_filter}
| | | | | | |    [lexpr{i'}
| | | | | | |    ;{k:Knd| k ∈ dom(da) }
| | | | | | |    ;λk.has-src(source(l);k)
| | | | | | |    ;IdLnk×Id×Type
| | | | | | |    ;λk.da-outlink-f(da;k)
| | | | | | |    ;fpf-dom-list(da)
| | | | | | |    ;<l,tg,da(rcv(l; tg))>]
| | | | | | |    THENA
| | | | | | |    (Reduce 0 THEN DVar `k' THEN DVar `k'
| | | | | | |     THEN
| | | | | | |     RWO
| | | | | | |     Thm* ∀i:Id, k:Knd.
| | | | | | |          has-src(i;k) ⟺ isrcv(k) & source(lnk(k)) = i
| | | | | | |     {assert_DASH_has_DASH_src}
| | | | | | |     -1
| | | | | | |     THEN
| | | | | | |     ExRepD)
| | | | | | |    THEN
| | | | | | |    Analyze -1
| | | | | | |    THEN
| | | | | | |    Thin -2
| | | | | | |    THEN
| | | | | | |    Analyze -1
| | | | | | |    THEN
| | | | | | |    Try Trivial
| | | | | | |    THEN
| | | | | | |    Reduce 0
| | | | | | |    THEN
| | | | | | |    InstConcl [rcv(l; tg)]
| | | | | | |
| | | | | | |\
| | | | | | | .....wf .....
| | | | | | | 1. D : Dsys
```

```
| | | | | | 2. L : Id List
| | | | | | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | | | | 4. ∀i:Id. Feasible(M(i))
| | | | | | 5. ∀i:Id.
| | | | | |    (i ∈ L)
| | | | | |    ⇒
| | | | | |    (∀ltg:(IdLnk×Id×Type).
| | | | | |     (ltg ∈ ma-outlinks(M(i);i))
| | | | | |     ⇒
| | | | | |     (destination(1of(ltg)) ∈ L)
| | | | | |     ⇒
| | | | | |     interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(
| | | | | |                                                 ltg))))
| | | | | | 6. l : IdLnk
| | | | | | 7. tg : Id
| | | | | | 8. (source(l) ∈ L)
| | | | | | 9. (destination(l) ∈ L)
| | | | | | 10. ds : x:Id fp-> Type
| | | | | | 11. da : a:Knd fp-> Type
| | | | | | 12. init : x:Id fp-> ds(x)?Void
| | | | | | 13. pre : a:Id fp-> State(ds)→da(locl(a))?Top→Prop
| | | | | | 14. ef :
| | | | | |     kx:Knd×Id fp-> State(ds)→da(1of(kx))?Top→
| | | | | |     ds(2of(kx))?Void
| | | | | | 15. send :
| | | | | |     kl:Knd×IdLnk fp-> (tg:Id
| | | | | |                        ×State(ds)→da(1of(kl))?Top→
| | | | | |                        (da(rcv(2of(kl); tg
| | | | | |                                ))?Void List)) List
| | | | | | 16. frame : x:Id fp-> Knd List
| | | | | | 17. sframe : ltg:IdLnk×Id fp-> Knd List
| | | | | | 18. M8 : Top
| | | | | | 19. M(source(l))
| | | | | |     =
| | | | | |     <ds,da,init,pre,ef,send,frame,sframe,M8>
| | | | | | 20. rcv(l; tg) ∈ dom(da)
| | | | | | ⊢  rcv(l; tg) ∈ {k:Knd| k ∈ dom(da) }  by Analyze
| | | | | |
| | | | | |\
| | | | | | 1. D : Dsys
| | | | | | 2. L : Id List
| | | | | | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | | | | 4. ∀i:Id. Feasible(M(i))
| | | | | | 5. ∀i:Id.
| | | | | | |  (i ∈ L)
| | | | | | |  ⇒
| | | | | | |  (∀ltg:(IdLnk×Id×Type).
| | | | | | |   (ltg ∈ ma-outlinks(M(i);i))
| | | | | | |   ⇒
| | | | | | |   (destination(1of(ltg)) ∈ L)
| | | | | | |   ⇒
| | | | | | |   interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(
| | | | | | |                                                 ltg))))
| | | | | | 6. l : IdLnk
| | | | | | 7. tg : Id
```

```
| | | | | | 8. (source(l) ∈ L)
| | | | | | 9. (destination(l) ∈ L)
| | | | | | 10. ds : x:Id fp-> Type
| | | | | | 11. da : a:Knd fp-> Type
| | | | | | 12. init : x:Id fp-> ds(x)?Void
| | | | | | 13. pre : a:Id fp-> State(ds)→da(locl(a))?Top→Prop
| | | | | | 14. ef :
| | | | | | |   kx:Knd×Id fp-> State(ds)→da(1of(kx))?Top→
| | | | | | |   ds(2of(kx))?Void
| | | | | | 15. send :
| | | | | | |   kl:Knd×IdLnk fp-> (tg:Id
| | | | | | |                      ×State(ds)→da(1of(kl))?Top→
| | | | | | |                      (da(rcv(2of(kl); tg
| | | | | | |                              ))?Void List)) List
| | | | | | 16. frame : x:Id fp-> Knd List
| | | | | | 17. sframe : ltg:IdLnk×Id fp-> Knd List
| | | | | | 18. M8 : Top
| | | | | | 19. M(source(l))
| | | | | | |   =
| | | | | | |   <ds,da,init,pre,ef,send,frame,sframe,M8>
| | | | | | 20. rcv(l; tg) ∈ dom(da)
| | | | | | ⊢  (rcv(l; tg) ∈ fpf-dom-list(da))
| | | | | | | by AssertBY
| | | | | | |    (fpf-dom-list(da) ∈ {k:Knd| k ∈ dom(da) } List)
| | | | | | |    Auto
| | | | | | |    THEN
| | | | | | |    DupHyp -2
| | | | | | |    THEN
| | | | | | |    Unfold `fpf-dom` -1
| | | | | | |    THEN
| | | | | | |    Fold `fpf-dom-list` -1
| | | | | | |    THEN
| | | | | | |    RWO
| | | | | | |    Thm* ∀eq:EqDecider(A), L:A List, x:A.
| | | | | | |         deq-member(eq;x;L) ⟺ (x ∈ L)
| | | | | | |    {assert_DASH_deq_DASH_member}
| | | | | | |    -1
| | | | | | |
| | | | | | |\
| | | | | | | .....wf .....
| | | | | | | 1. D : Dsys
| | | | | | | 2. L : Id List
| | | | | | | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | | | | | 4. ∀i:Id. Feasible(M(i))
| | | | | | | 5. ∀i:Id.
| | | | | | |    (i ∈ L)
| | | | | | |    ⇒
| | | | | | |    (∀ltg:(IdLnk×Id×Type).
| | | | | | |     (ltg ∈ ma-outlinks(M(i);i))
| | | | | | |     ⇒
| | | | | | |     (destination(1of(ltg)) ∈ L)
| | | | | | |     ⇒
| | | | | | |     interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(
| | | | | | |                                                 ltg))))
| | | | | | | 6. l : IdLnk
```

```
| | | | | | | 7. tg : Id
| | | | | | | 8. (source(l) ∈ L)
| | | | | | | 9. (destination(l) ∈ L)
| | | | | | | 10. ds : x:Id fp-> Type
| | | | | | | 11. da : a:Knd fp-> Type
| | | | | | | 12. init : x:Id fp-> ds(x)?Void
| | | | | | | 13. pre : a:Id fp-> State(ds)→da(locl(a))?Top→Prop
| | | | | | | 14. ef :
| | | | | | |     kx:Knd×Id fp-> State(ds)→da(1of(kx))?Top→
| | | | | | |     ds(2of(kx))?Void
| | | | | | | 15. send :
| | | | | | |     kl:Knd×IdLnk fp-> (tg:Id
| | | | | | |                        ×State(ds)→da(1of(kl))?Top→
| | | | | | |                        (da(rcv(2of(kl); tg
| | | | | | |                                ))?Void List)) List
| | | | | | | 16. frame : x:Id fp-> Knd List
| | | | | | | 17. sframe : ltg:IdLnk×Id fp-> Knd List

| | | | | | | 18. M8 : Top
| | | | | | | 19. M(source(l))
| | | | | | |     =
| | | | | | |     <ds,da,init,pre,ef,send,frame,sframe,M8>
| | | | | | | 20. rcv(l; tg) ∈ dom(da)
| | | | | | | 21. fpf-dom-list(da) ∈ {k:Knd| k ∈ dom(da) } List
| | | | | | | 22. deq-member(KindDeq;rcv(l; tg);fpf-dom-list(da))
| | | | | | | ⊢ fpf-dom-list(da) ∈ Knd List
| | | | | | |   by DoSubsume THEN UseSubtypeRelLemmas THEN Analyze 0
| | | | | | |
| | | | | | .\
| | | | | | . 1. D : Dsys
| | | | | | . 2. L : Id List
| | | | | | . 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | | | | . 4. ∀i:Id. Feasible(M(i))
| | | | | | . 5. ∀i:Id.
| | | | | | . |  (i ∈ L)
| | | | | | . |  ⇒
| | | | | | . |  (∀ltg:(IdLnk×Id×Type).
| | | | | | . |   (ltg ∈ ma-outlinks(M(i);i))
| | | | | | . |   ⇒
| | | | | | . |   (destination(1of(ltg)) ∈ L)
| | | | | | . |   ⇒
| | | | | | . |   interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(
| | | | | | . |                                                ltg))))
| | | | | | . 6. l : IdLnk
| | | | | | . 7. tg : Id
| | | | | | . 8. (source(l) ∈ L)
| | | | | | . 9. (destination(l) ∈ L)
| | | | | | . 10. ds : x:Id fp-> Type
| | | | | | . 11. da : a:Knd fp-> Type
| | | | | | . 12. init : x:Id fp-> ds(x)?Void
| | | | | | . 13. pre : a:Id fp-> State(ds)→da(locl(a))?Top→Prop
| | | | | | . 14. ef :
| | | | | | . |   kx:Knd×Id fp-> State(ds)→da(1of(kx))?Top→
| | | | | | . |   ds(2of(kx))?Void
| | | | | | . 15. send :
| | | | | | . |   kl:Knd×IdLnk fp-> (tg:Id
```

```
| | | | | | . |                       ×State(ds)→da(1of(kl))?Top→
| | | | | | . |                       (da(rcv(2of(kl); tg
| | | | | | . |                               ))?Void List)) List
| | | | | | . 16. frame : x:Id fp-> Knd List
| | | | | | . 17. sframe : ltg:IdLnk×Id fp-> Knd List
| | | | | | . 18. M8 : Top
| | | | | | . 19. M(source(l))
| | | | | | . |   =
| | | | | | . |   <ds,da,init,pre,ef,send,frame,sframe,M8>
| | | | | | . 20. rcv(l; tg) ∈ dom(da)
| | | | | | . 21. fpf-dom-list(da) ∈ {k:Knd| k ∈ dom(da) } List
| | | | | | . 22. (rcv(l; tg) ∈ fpf-dom-list(da))
| | | | | | . ⊢  (rcv(l; tg) ∈ fpf-dom-list(da))
| | | | | | . | by RepeatFor 2 ParallelLast THEN ExRepD
| | | | | | . |
| | | | | | . |\
| | | | | | . | 1. D : Dsys
| | | | | | . | 2. L : Id List
| | | | | | . | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | | | | . | 4. ∀i:Id. Feasible(M(i))
| | | | | | . | 5. ∀i:Id.
| | | | | | . |    (i ∈ L)
| | | | | | . |    ⇒
| | | | | | . |    (∀ltg:(IdLnk×Id×Type).
| | | | | | . |     (ltg ∈ ma-outlinks(M(i);i))
| | | | | | . |     ⇒
| | | | | | . |     (destination(1of(ltg)) ∈ L)
| | | | | | . |     ⇒
| | | | | | . |     interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(
| | | | | | . |                                                 ltg))))
| | | | | | . | 6. l : IdLnk
| | | | | | . | 7. tg : Id
| | | | | | . | 8. (source(l) ∈ L)
| | | | | | . | 9. (destination(l) ∈ L)
| | | | | | . | 10. ds : x:Id fp-> Type
| | | | | | . | 11. da : a:Knd fp-> Type
| | | | | | . | 12. init : x:Id fp-> ds(x)?Void
| | | | | | . | 13. pre : a:Id fp-> State(ds)→da(locl(a))?Top→Prop
| | | | | | . | 14. ef :
| | | | | | . |     kx:Knd×Id fp-> State(ds)→da(1of(kx))?Top→
| | | | | | . |     ds(2of(kx))?Void
| | | | | | . | 15. send :
| | | | | | . |     kl:Knd×IdLnk fp-> (tg:Id
| | | | | | . |                        ×State(ds)→da(1of(kl))?Top→
| | | | | | . |                        (da(rcv(2of(kl); tg
| | | | | | . |                                ))?Void List)) List
| | | | | | . | 16. frame : x:Id fp-> Knd List
| | | | | | . | 17. sframe : ltg:IdLnk×Id fp-> Knd List
| | | | | | . | 18. M8 : Top
| | | | | | . | 19. M(source(l))
| | | | | | . |     =
| | | | | | . |     <ds,da,init,pre,ef,send,frame,sframe,M8>
| | | | | | . | 20. rcv(l; tg) ∈ dom(da)
| | | | | | . | 21. fpf-dom-list(da) ∈ {k:Knd| k ∈ dom(da) } List
| | | | | | . | 22. i : ℕ
```

```
| | | | | | . | 23. i<||fpf-dom-list(da)||
| | | | | | . | 24. rcv(l; tg) = fpf-dom-list(da)[i]
| | | | | | . | ⊢  rcv(l; tg) = fpf-dom-list(da)[i]  by Analyze
| | | | | | . |
| | | | | | . .\
| | | | | | . . 1. D : Dsys
| | | | | | . . 2. L : Id List
| | | | | | . . 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | | | | . . 4. ∀i:Id. Feasible(M(i))
| | | | | | . . 5. ∀i:Id.
| | | | | | . .    (i ∈ L)
| | | | | | . .    ⇒
| | | | | | . .    (∀ltg:(IdLnk×Id×Type).
| | | | | | . .     (ltg ∈ ma-outlinks(M(i);i))
| | | | | | . .     ⇒
| | | | | | . .     (destination(1of(ltg)) ∈ L)
| | | | | | . .     ⇒
| | | | | | . .     interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(
| | | | | | . .                                              ltg))))
| | | | | | . . 6. l : IdLnk
| | | | | | . . 7. tg : Id
| | | | | | . . 8. (source(l) ∈ L)
| | | | | | . . 9. (destination(l) ∈ L)
| | | | | | . . 10. ds : x:Id fp-> Type
| | | | | | . . 11. da : a:Knd fp-> Type
| | | | | | . . 12. init : x:Id fp-> ds(x)?Void
| | | | | | . . 13. pre : a:Id fp-> State(ds)→da(locl(a))?Top→Prop
| | | | | | . . 14. ef :
| | | | | | . .     kx:Knd×Id fp-> State(ds)→da(1of(kx))?Top→
| | | | | | . .     ds(2of(kx))?Void
| | | | | | . . 15. send :
| | | | | | . .     kl:Knd×IdLnk fp-> (tg:Id
| | | | | | . .                        ×State(ds)→da(1of(kl))?Top→
| | | | | | . .                        (da(rcv(2of(kl); tg
| | | | | | . .                                ))?Void List)) List
| | | | | | . . 16. frame : x:Id fp-> Knd List
| | | | | | . . 17. sframe : ltg:IdLnk×Id fp-> Knd List
| | | | | | . . 18. M8 : Top
| | | | | | . . 19. M(source(l))
| | | | | | . .     =
| | | | | | . .     <ds,da,init,pre,ef,send,frame,sframe,M8>
| | | | | | . . 20. rcv(l; tg) ∈ dom(da)
| | | | | | . . 21. fpf-dom-list(da) ∈ {k:Knd| k ∈ dom(da) } List
| | | | | | . . 22. i : ℕ
| | | | | | . . 23. i<||fpf-dom-list(da)||
| | | | | | . . 24. rcv(l; tg) = fpf-dom-list(da)[i]
| | | | | | . . 25. i1 : ℕ
| | | | | | . . 26. i1<||fpf-dom-list(da)||
| | | | | | . . ⊢  rcv(l; tg) ∈ {k:Knd| k ∈ dom(da) }  by Analyze
| | | | | | .
| | | | | |\
| | | | | | 1. D : Dsys
| | | | | | 2. L : Id List
| | | | | | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | | | | 4. ∀i:Id. Feasible(M(i))
```

```
| | | | | | 5. ∀i:Id.
| | | | | |    (i ∈ L)
| | | | | |    ⇒
| | | | | |    (∀ltg:(IdLnk×Id×Type).
| | | | | |     (ltg ∈ ma-outlinks(M(i);i))
| | | | | |     ⇒
| | | | | |     (destination(1of(ltg)) ∈ L)
| | | | | |     ⇒
| | | | | |     interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(
| | | | | |                                                 ltg))))
| | | | | | 6. l : IdLnk
| | | | | | 7. tg : Id
| | | | | | 8. (source(l) ∈ L)
| | | | | | 9. (destination(l) ∈ L)
| | | | | | 10. ds : x:Id fp-> Type
| | | | | | 11. da : a:Knd fp-> Type
| | | | | | 12. init : x:Id fp-> ds(x)?Void
| | | | | | 13. pre : a:Id fp-> State(ds)→da(locl(a))?Top→Prop
| | | | | | 14. ef :
| | | | | |     kx:Knd×Id fp-> State(ds)→da(1of(kx))?Top→
| | | | | |     ds(2of(kx))?Void
| | | | | | 15. send :
| | | | | |     kl:Knd×IdLnk fp-> (tg:Id
| | | | | |                        ×State(ds)→da(1of(kl))?Top→
| | | | | |                        (da(rcv(2of(kl); tg
| | | | | |                                ))?Void List)) List
| | | | | | 16. frame : x:Id fp-> Knd List
| | | | | | 17. sframe : ltg:IdLnk×Id fp-> Knd List
| | | | | | 18. M8 : Top
| | | | | | 19. M(source(l))
| | | | | |     =
| | | | | |     <ds,da,init,pre,ef,send,frame,sframe,M8>
| | | | | | 20. rcv(l; tg) ∈ dom(da)
| | | | | | ⊢  has-src(source(l);rcv(l; tg))
| | | | | |   by Unfold ‘has-src‘ 0
| | | | | |      THEN
| | | | | |      Unfolds [‘isrcv‘;‘rcv‘;‘lnk‘] 0
| | | | | |      THEN
| | | | | |      Reduce 0
| | | | | |      THEN
| | | | | |      RW assert_pushdownC 0
| | | | | |
| | | | | |\
| | | | | | 1. D : Dsys
| | | | | | 2. L : Id List
| | | | | | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | | | | 4. ∀i:Id. Feasible(M(i))
| | | | | | 5. ∀i:Id.
| | | | | |    (i ∈ L)
| | | | | |    ⇒
| | | | | |    (∀ltg:(IdLnk×Id×Type).
| | | | | |     (ltg ∈ ma-outlinks(M(i);i))
| | | | | |     ⇒
| | | | | |     (destination(1of(ltg)) ∈ L)
| | | | | |     ⇒
```

```
| | | | | | |     interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(
| | | | | | |                                                 ltg))))
| | | | | | | 6. l : IdLnk
| | | | | | | 7. tg : Id
| | | | | | | 8. (source(l) ∈ L)
| | | | | | | 9. (destination(l) ∈ L)
| | | | | | | 10. ds : x:Id fp-> Type
| | | | | | | 11. da : a:Knd fp-> Type
| | | | | | | 12. init : x:Id fp-> ds(x)?Void
| | | | | | | 13. pre : a:Id fp-> State(ds)→da(locl(a))?Top→Prop
| | | | | | | 14. ef :
| | | | | | |     kx:Knd×Id fp-> State(ds)→da(1of(kx))?Top→
| | | | | | |     ds(2of(kx))?Void
| | | | | | | 15. send :
| | | | | | |     kl:Knd×IdLnk fp-> (tg:Id
| | | | | | |                        ×State(ds)→da(1of(kl))?Top→
| | | | | | |                        (da(rcv(2of(kl); tg
| | | | | | |                                ))?Void List)) List
| | | | | | | 16. frame : x:Id fp-> Knd List
| | | | | | | 17. sframe : ltg:IdLnk×Id fp-> Knd List
| | | | | | | 18. M8 : Top
| | | | | | | 19. M(source(l))
| | | | | | |     =
| | | | | | |     <ds,da,init,pre,ef,send,frame,sframe,M8>
| | | | | | | 20. rcv(l; tg) ∈ dom(da)
| | | | | | | ⊢  <l,tg,da(rcv(l; tg))> = da-outlink-f(da;rcv(l; tg))
| | | | | | |   by Repeat
| | | | | | |      (Unfolds [‘da-outlink-f‘;‘lnk‘;‘rcv‘;‘tagof‘] 0
| | | | | | |       THEN
| | | | | | |       Reduce 0)
| | | | | | |      THEN
| | | | | | |      Fold ‘rcv‘ 0
| | | | | | |
| | | | | .\
| | | | | . .....subterm .....
| | | | | . 1. D : Dsys
| | | | | . 2. L : Id List
| | | | | . 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | | | . 4. ∀i:Id. Feasible(M(i))
| | | | | . 5. ∀i:Id.
| | | | | .    (i ∈ L)
| | | | | .    ⇒
| | | | | .    (∀ltg:(IdLnk×Id×Type).
| | | | | .     (ltg ∈ ma-outlinks(M(i);i))
| | | | | .     ⇒
| | | | | .     (destination(1of(ltg)) ∈ L)
| | | | | .     ⇒
| | | | | .     interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(
| | | | | .                                                 ltg))))
| | | | | . 6. l : IdLnk
| | | | | . 7. tg : Id
| | | | | . 8. (source(l) ∈ L)
| | | | | . 9. (destination(l) ∈ L)
| | | | | . 10. ds : x:Id fp-> Type
| | | | | . 11. da : a:Knd fp-> Type
```

```
| | | | | . 12. init : x:Id fp-> ds(x)?Void
| | | | | . 13. pre : a:Id fp-> State(ds)→da(locl(a))?Top→Prop
| | | | | . 14. ef :
| | | | | .     kx:Knd×Id fp-> State(ds)→da(1of(kx))?Top→
| | | | | .     ds(2of(kx))?Void
| | | | | . 15. send :
| | | | | .     kl:Knd×IdLnk fp-> (tg:Id
| | | | | .                        ×State(ds)→da(1of(kl))?Top→
| | | | | .                        (da(rcv(2of(kl); tg
| | | | | .                                ))?Void List)) List
| | | | | . 16. frame : x:Id fp-> Knd List
| | | | | . 17. sframe : ltg:IdLnk×Id fp-> Knd List
| | | | | . 18. M8 : Top
| | | | | . 19. M(source(l))
| | | | | .     =
| | | | | .     <ds,da,init,pre,ef,send,frame,sframe,M8>
| | | | | . 20. rcv(l; tg) ∈ dom(da)
| | | | | . 21. y : {k:Knd| k ∈ dom(da) }
| | | | | . 22. has-src(source(l);y)
| | | | | . ⊢  y ∈ {k:Knd| k ∈ dom(da) & isrcv(k) }
| | | | | .   by RWO
| | | | | .      Thm* ∀i:Id, k:Knd.
| | | | | .           has-src(i;k) ⟺ isrcv(k) & source(lnk(k)) = i
| | | | | .      {assert_DASH_has_DASH_src}
| | | | | .      -1
| | | | | .      THEN
| | | | | .      ExRepD
| | | | |
| | | | .\
| | | | . 1. D : Dsys
| | | | . 2. L : Id List
| | | | . 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | | . 4. ∀i:Id. Feasible(M(i))
| | | | . 5. ∀i:Id.
| | | | .    (i ∈ L)
| | | | .    ⇒
| | | | .    (∀ltg:(IdLnk×Id×Type).
| | | | .     (ltg ∈ ma-outlinks(M(i);i))
| | | | .     ⇒
| | | | .     (destination(1of(ltg)) ∈ L)
| | | | .     ⇒
| | | | .     interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
| | | | . 6. l : IdLnk
| | | | . 7. tg : Id
| | | | . 8. (source(l) ∈ L)
| | | | . 9. (destination(l) ∈ L)
| | | | . 10. rcv(l; tg) ∈ dom(1of(2of(M(source(l)))))
| | | | . 11. interface-check(D;l;tg;1of(2of(M(source(l))))(rcv
| | | | .     (l; tg)))
| | | | . ⊢  1of(2of(M(source(l))))(rcv
| | | | .    (l; tg)) ⊆r M(destination(l)).din(l,tg)
| | | | .   by Unfold ‘interface-check‘ -1
| | | |
| | | .\
| | | . .....falsecase .....
```

```
| | | . 1. D : Dsys
| | | . 2. L : Id List
| | | . 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | | . 4. ∀i:Id. Feasible(M(i))
| | | . 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
| | | .                                    ⇒
| | | .    interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
| | | . 6. l : IdLnk
| | | . 7. tg : Id
| | | . 8. (source(l) ∈ L)
| | | . 9. (destination(l) ∈ L)
| | | . 10. ¬₂rcv(l; tg) ∈ dom(1of(2of(M(source(l)))))
| | | . ⊢  Void ⊆r M(destination(l)).din(l,tg)
| | | .   by At Type (Analyze 0)
| | |
| | .\
| | . 1. D : Dsys
| | . 2. L : Id List
| | . 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | . 4. ∀i:Id. Feasible(M(i))
| | . 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
| | . |                                    ⇒
| | . |  interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
| | . 6. l : IdLnk
| | . 7. tg : Id
| | . 8. (source(l) ∈ L)
| | . 9. ¬(destination(l) ∈ L)
| | . ⊢  M(source(l)).dout(l,tg) ⊆r M(destination(l)).din(l,tg)
| | . | by Subst (M(destination(l)) = ) 0
| | . |
| | . |\
| | . | .....equality .....
| | . | 1. D : Dsys
| | . | 2. L : Id List
| | . | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | . | 4. ∀i:Id. Feasible(M(i))
| | . | 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
| | . | |                                    ⇒
| | . | |  interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
| | . | 6. l : IdLnk
| | . | 7. Id
| | . | 8. (source(l) ∈ L)
| | . | 9. ¬(destination(l) ∈ L)
| | . | ⊢  M(destination(l)) =
| | . | | by InstHyp [destination(l)] 3 THEN Analyze -1
| | . | |
| | . | 1. D : Dsys
| | . | 2. L : Id List
| | . | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | . | 4. ∀i:Id. Feasible(M(i))
| | . | 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
| | . |                                    ⇒
| | . |    interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
| | . | 6. l : IdLnk
| | . | 7. Id
```

```
| | . | 8. (source(l) ∈ L)
| | . | 9. ¬(destination(l) ∈ L)
| | . | 10. ma-is-empty(M(destination(l)))
| | . | ⊢  M(destination(l)) =
| | . |   by RWO
| | . |      Thm* ∀M:MsgA. ma-is-empty(M) ⟺ M =  ∈ MsgA
| | . |      {assert_DASH_ma_DASH_is_DASH_empty}
| | . |      -1
| | . |
| | . .\
| | . . 1. D : Dsys
| | . . 2. L : Id List
| | . . 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| | . . 4. ∀i:Id. Feasible(M(i))
| | . . 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
| | . .                                    ⇒
| | . .    interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg)))))
| | . . 6. l : IdLnk
| | . . 7. tg : Id
| | . . 8. (source(l) ∈ L)
| | . . 9. ¬(destination(l) ∈ L)
| | . . ⊢  M(source(l)).dout(l,tg) ⊆r .din(l,tg)
| | . .   by Repeat
| | . .      (Unfolds [‘ma-din‘;‘ma-empty‘;‘mk-ma‘] 0 THEN Reduce 0)
| | . .      THEN
| | . .      At Type (Analyze 0)
| |
| .\
| . 1. D : Dsys
| . 2. L : Id List
| . 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| . 4. ∀i:Id. Feasible(M(i))
| . 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
| . |                                  ⇒
| . |  interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg)))))
| . 6. l : IdLnk
| . 7. tg : Id
| . 8. ¬(source(l) ∈ L)
| . ⊢  M(source(l)).dout(l,tg) ⊆r M(destination(l)).din(l,tg)
| . | by Subst (M(source(l)) = ) 0
| . |
| . |\
| . | .....equality .....
| . | 1. D : Dsys
| . | 2. L : Id List
| . | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| . | 4. ∀i:Id. Feasible(M(i))
| . | 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
| . | |                                  ⇒
| . | |  interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg)))))
| . | 6. l : IdLnk
| . | 7. Id
| . | 8. ¬(source(l) ∈ L)
| . | ⊢  M(source(l)) =   by InstHyp [source(l)] 3 THEN Analyze -1
| . | |
```

```
| . | 1. D : Dsys
| . | 2. L : Id List
| . | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| . | 4. ∀i:Id. Feasible(M(i))
| . | 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
| . |                                    ⇒
| . |    interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
| . | 6. l : IdLnk
| . | 7. Id
| . | 8. ¬(source(l) ∈ L)
| . | 9. ma-is-empty(M(source(l)))
| . | ⊢  M(source(l)) =
| . |   by RWO
| . |      Thm* ∀M:MsgA. ma-is-empty(M) ⟺ M =  ∈ MsgA
| . |      {assert_DASH_ma_DASH_is_DASH_empty}
| . |      -1
| . |
| . .\
| . . 1. D : Dsys
| . . 2. L : Id List
| . . 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
| . . 4. ∀i:Id. Feasible(M(i))
| . . 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
| . .                                    ⇒
| . .    interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
| . . 6. l : IdLnk
| . . 7. tg : Id
| . . 8. ¬(source(l) ∈ L)
| . . ⊢  .dout(l,tg) ⊆r M(destination(l)).din(l,tg)
| . .   by Repeat
| . .      (Unfolds [‘ma-dout‘;‘ma-empty‘;‘mk-ma‘] 0 THEN Reduce 0)
| . .      THEN
| . .      At Type (Analyze 0)
|
.\
. 1. D : Dsys
. 2. L : Id List
. 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
. 4. ∀i:Id. Feasible(M(i))
. 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
. |                                  ⇒
. |  interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
. ⊢  ∀i:Id.
. |  finite-type(...)
. | by Auto
. |    THEN
. |    BackThru
. |    Thm* ∀P:(T→Prop).
. |         (∀x:T. Dec(P(x)))
. |         ⇒
. |         (finite-type({x:T| P(x) })
. |          ⟺
. |          (∃L:T List. ∀x:T. P(x) ⇒ (x ∈ L)))
. |    {finite_DASH_decidable_DASH_set}
. |
```

```
. 1. D : Dsys
. 2. L : Id List
. 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
. 4. ∀i:Id. Feasible(M(i))
. 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
. |                                    ⇒
. |  interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
. 6. i : Id
. ⊢  ∃L:IdLnk List.
. |  ∀l:IdLnk.
. |  destination(l) = i & M(source(l)) sends on link l ⇒ (l ∈ L)
. | by Assert
. |    (∀i:Id.
. |     map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(i))))))))
. |     ∈ IdLnk List)
. |
. |\
. | .....assertion .....
. | 1. D : Dsys
. | 2. L : Id List
. | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
. | 4. ∀i:Id. Feasible(M(i))
. | 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
. |                                    ⇒
. |    interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
. | 6. Id
. | ⊢  ∀i:Id.
. |    map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(i))))))))
. |    ∈ IdLnk List
. |   by Auto THEN GenConclMsgA THEN Thin -1 THEN Unfold_MsgA -1
. |      THEN
. |      DVar 'send'
. |      THEN
. |      Reduce 0
. |
. .\
. . 1. D : Dsys
. . 2. L : Id List
. . 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
. . 4. ∀i:Id. Feasible(M(i))
. . 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
. . |                                    ⇒
. . |  interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
. . 6. i : Id
. . 7. ∀i:Id.
. . |  map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(i))))))))
. . |  ∈ IdLnk List
. . ⊢  ∃L:IdLnk List.
. . |  ∀l:IdLnk.
. . |  destination(l) = i & M(source(l)) sends on link l ⇒ (l ∈ L)
. . | by AssertBY
. . |    ((λi.map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(i))))))))))
. . |     ∈ Id→(IdLnk List))
. . |    (Auto THEN BackThruSomeHyp)
. . |    THEN
```

```
. . |    InstConcl
. . |    [concat(map(λi.map(λp.
. . |     2of(p);1of(1of(2of(2of(2of(2of(2of(2of(M(i)))))))));L))]
. . |
. . 1. D : Dsys
. . 2. L : Id List
. . 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
. . 4. ∀i:Id. Feasible(M(i))
. . 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
. . |                                    ⇒
. . |  interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
. . 6. i : Id
. . 7. ∀i:Id.
. . |  map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(i))))))))
. . |  ∈ IdLnk List
. . 8. (λi.map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(i)))))))))
. . |  ∈ Id→(IdLnk List)
. . 9. l : IdLnk
. . 10. destination(l) = i
. . 11. M(source(l)) sends on link l
. . ⊢  (l ∈ concat(map(λi.map(λp.
. . |  2of(p);1of(1of(2of(2of(2of(2of(2of(M(i)))))));L)))
. . | by RWO
. . |    Thm* ∀ll:(T List) List, x:T.
. . |         (x ∈ concat(ll)) ⟺ (∃l:T List. (l ∈ ll) & (x ∈ l))
. . |    {member_DASH_concat}
. . |    0
. . |
. . 1. D : Dsys
. . 2. L : Id List
. . 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
. . 4. ∀i:Id. Feasible(M(i))
. . 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
. . |                                    ⇒
. . |  interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
. . 6. i : Id
. . 7. ∀i:Id.
. . |  map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(i)))))))))
. . |  ∈ IdLnk List
. . 8. (λi.map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(i)))))))))
. . |  ∈ Id→(IdLnk List)
. . 9. l : IdLnk
. . 10. destination(l) = i
. . 11. M(source(l)) sends on link l
. . ⊢  ∃l1:IdLnk List.
. . |  (l1 ∈ map(λi.map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(
. . |                                    M(i)))))))));L))
. . |  & (l ∈ l1)  by InstHyp [source(l)] 3 THEN Analyze -1
. . |
. . |\
. . | 1. D : Dsys
. . | 2. L : Id List
. . | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
. . | 4. ∀i:Id. Feasible(M(i))
. . | 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
```

```
. . | |                                        ⇒
. . | |  interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
. . | 6. i : Id
. . | 7. ∀i:Id.
. . | |  map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(i))))))))
. . | |  ∈ IdLnk List
. . | 8. (λi.map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(i))))))))))
. . | |  ∈ Id→(IdLnk List)
. . | 9. l : IdLnk
. . | 10. destination(l) = i
. . | 11. M(source(l)) sends on link l
. . | 12. (source(l) ∈ L)
. . | ⊢  ∃l1:IdLnk List.
. . | |  (l1 ∈ map(λi.map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(
. . | |                                        M(i))))))))));L))
. . | |  & (l ∈ l1)
. . | | by RWO
. . | |    Thm* ∀a:T List, x:T', f:(T→T').
. . | |         (x ∈ map(f;a)) ⟺ (∃y:T. (y ∈ a) & x = f(y))
. . | |    {member_map}
. . | |    0
. . | |    THEN
. . | |    Reduce 0
. . | |    THEN
. . | |    InstConcl
. . | |    [map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(
. . | |                              M(source(l))))))))))]
. . | |    THEN
. . | |    Try BackThruSomeHyp
. . | |
. . | |\
. . | | 1. D : Dsys
. . | | 2. L : Id List
. . | | 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
. . | | 4. ∀i:Id. Feasible(M(i))
. . | | 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
. . | |                                        ⇒
. . | |    interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
. . | | 6. i : Id
. . | | 7. ∀i:Id.
. . | |    map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(i))))))))
. . | |    ∈ IdLnk List
. . | | 8. (λi.map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(i))))))))))
. . | |    ∈ Id→(IdLnk List)
. . | | 9. l : IdLnk
. . | | 10. destination(l) = i
. . | | 11. M(source(l)) sends on link l
. . | | 12. (source(l) ∈ L)
. . | | ⊢  ∃y:Id.
. . | |    (y ∈ L)
. . | |    & map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(
. . | |                              M(source(l))))))))))
. . | |      =
. . | |      map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(y))))))))
. . | |   by InstConcl [source(l)] THEN Try (Fold ‘member‘ 0)
```

```
. . | |      THEN
. . | |      Try BackThruSomeHyp
. . | |
. . | .\
. . | . 1. D : Dsys
. . | . 2. L : Id List
. . | . 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
. . | . 4. ∀i:Id. Feasible(M(i))
. . | . 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
. . | .                                    ⇒
. . | .    interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
. . | . 6. i : Id
. . | . 7. ∀i:Id.
. . | .    map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(2of(M(i))))))))))
. . | .    ∈ IdLnk List
. . | . 8. (λi.map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(i)))))))))))
. . | .    ∈ Id→(IdLnk List)
. . | . 9. l : IdLnk
. . | . 10. destination(l) = i
. . | . 11. M(source(l)) sends on link l
. . | . 12. (source(l) ∈ L)
. . | . ⊢  (l ∈ map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(
. . | .                                 M(source(l))))))))))
. . | .   by Unfold ‘ma-sends-on‘ -2
. . | .      THEN
. . | .      RWO
. . | .      Thm* ∀eq:EqDecider(A), L:A List, x:A.
. . | .           deq-member(eq;x;L) ⟺ (x ∈ L)
. . | .      {assert_DASH_deq_DASH_member}
. . | .      -2
. . | .      THEN
. . | .      Try BackThruSomeHyp
. . |
. . .\
. . . 1. D : Dsys
. . . 2. L : Id List
. . . 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
. . . 4. ∀i:Id. Feasible(M(i))
. . . 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
. . . |                                  ⇒
. . . |  interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg))))
. . . 6. i : Id
. . . 7. ∀i:Id.
. . . |  map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(i)))))))))
. . . |  ∈ IdLnk List
. . . 8. (λi.map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(i))))))))))
. . . |  ∈ Id→(IdLnk List)
. . . 9. l : IdLnk
. . . 10. destination(l) = i
. . . 11. M(source(l)) sends on link l
. . . 12. ma-is-empty(M(source(l)))
. . . ⊢  ∃l1:IdLnk List.
. . . |  (l1 ∈ map(λi.map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(
. . . |                                      M(i))))))));L))
. . . |  & (l ∈ l1)
```

```
. . . | by RWO
. . . |    Thm* ∀M:MsgA. ma-is-empty(M) ⟺ M =  ∈ MsgA

. . . |    {assert_DASH_ma_DASH_is_DASH_empty}
. . . |    -1
. . . |    THEN
. . . |    HypSubst -1 -2
. . . |
. . . 1. D : Dsys
. . . 2. L : Id List
. . . 3. ∀i:Id. (i ∈ L) ∨ ma-is-empty(M(i))
. . . 4. ∀i:Id. Feasible(M(i))
. . . 5. (∀i∈L.(∀ltg∈ma-outlinks(M(i);i).(destination(1of(ltg)) ∈ L)
. . .                                    ⇒
. . .    interface-check(D;1of(ltg);1of(2of(ltg));2of(2of(ltg)))))
. . . 6. i : Id
. . . 7. ∀i:Id.
. . .    map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(i))))))))
. . .    ∈ IdLnk List
. . . 8. (λi.map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(M(i))))))))))
. . .    ∈ Id→(IdLnk List)
. . . 9. l : IdLnk
. . . 10. destination(l) = i
. . . 11.  sends on link l
. . . 12. M(source(l)) =
. . . ⊢  ∃l1:IdLnk List.
. . .    (l1 ∈ map(λi.map(λp.2of(p);1of(1of(2of(2of(2of(2of(2of(
. . .                                       M(i))))))))));L))
. . .    & (l ∈ l1)
. . .   by Repeat
. . .      (Unfolds [‘ma-sends-on‘;‘ma-empty‘;‘mk-ma‘;‘fpf-empty‘] -2
. . .       THEN
. . .       Reduce -2)
END
```