```
ma-send-sub  THM

⊢  ∀M1,M2:MsgA.
|  M1 ⊆ M2
|  ⇒
|  (∀k:Knd, l:IdLnk, s:M2.state, v:M2.da(k), i:Id,
|    ms:(tg:Id×if source(l) = i→ M2.da(rcv(l; tg)) else Top fi) List.
|   M2.send(k;l;s;v;ms;i) ⇒ M1.send(k;l;s;v;ms;i))
| by RepeatFor 9 (Analyze 0 THENA Complete Auto) THEN Unfold_MsgA -9
|    THEN
|    Unfold_MsgA -8
|    THEN
|    SplitAndHyps
|    THEN
|    Unfold ‘fpf-val‘ 0
|    THEN
|    SplitOnConclITE
|
|\
| .....truecase .....
| 1. ds : x:Id fp-> Type
| 2. da : a:Knd fp-> Type
| 3. init : x:Id fp-> ds(x)?Void
| 4. pre : a:Id fp-> State(ds)→da(locl(a))?Top→Prop
| 5. ef : kx:Knd×Id fp-> State(ds)→da(1of(kx))?Top→ds(2of(kx))?Void
| 6. send :
| |  kl:Knd×IdLnk fp-> (tg:Id
| |                     ×State(ds)→da(1of(kl))?Top→
| |                     (da(rcv(2of(kl); tg))?Void List)) List
| 7. frame : x:Id fp-> Knd List
| 8. sframe : ltg:IdLnk×Id fp-> Knd List
| 9. Top
| 10. d1 : x:Id fp-> Type
| 11. d2 : a:Knd fp-> Type
| 12. i1 : x:Id fp-> d1(x)?Void
| 13. p1 : a:Id fp-> State(d1)→d2(locl(a))?Top→Prop
| 14. e1 : kx:Knd×Id fp-> State(d1)→d2(1of(kx))?Top→d1(2of(kx))?Void
| 15. s1 :
| |   kl:Knd×IdLnk fp-> (tg:Id
| |                      ×State(d1)→d2(1of(kl))?Top→
| |                      (d2(rcv(2of(kl); tg))?Void List)) List
| 16. f1 : x:Id fp-> Knd List
| 17. s2 : ltg:IdLnk×Id fp-> Knd List
| 18. Top
| 19. ds ⊆ d1
| 20. da ⊆ d2
| 21. init ⊆ i1
| 22. pre ⊆ p1
| 23. ef ⊆ e1
| 24. send ⊆ s1
| 25. frame ⊆ f1
| 26. sframe ⊆ s2
| 27. k : Knd
| 28. l : IdLnk
| 29. s : State(d1)
```

```
| 30. v : d2(k)?Top
| 31. i : Id
| 32. ms :
| |   (tg:Id×if source(l) = i→ d2(rcv(l; tg))?Top else Top fi) List
| 33. source(l) = i
| ⊢  (<k,l> ∈ dom(s1)
| |   ⇒
| |   ms
| |   =
| |   concat(map(λtgf.map(λx.<1of(tgf),x>;2of(tgf)(s,v));s1(<k,l>))))
| |  ⇒
| |  <k,l> ∈ dom(send)
| |  ⇒
| |  ms
| |  =
| |  concat(map(λtgf.map(λx.<1of(tgf),x>;2of(tgf)(s,v));send(<k,l>)))
| | by SplitOnHypITE -2
| |
| |\
| | 1. ds : x:Id fp-> Type
| | 2. da : a:Knd fp-> Type
| | 3. init : x:Id fp-> ds(x)?Void
| | 4. pre : a:Id fp-> State(ds)→da(locl(a))?Top→Prop
| | 5. ef :
| |    kx:Knd×Id fp-> State(ds)→da(1of(kx))?Top→ds(2of(kx))?Void
| | 6. send :
| |    kl:Knd×IdLnk fp-> (tg:Id
| |                       ×State(ds)→da(1of(kl))?Top→
| |                       (da(rcv(2of(kl); tg))?Void List)) List
| | 7. frame : x:Id fp-> Knd List
| | 8. sframe : ltg:IdLnk×Id fp-> Knd List
| | 9. Top
| | 10. d1 : x:Id fp-> Type
| | 11. d2 : a:Knd fp-> Type
| | 12. i1 : x:Id fp-> d1(x)?Void
| | 13. p1 : a:Id fp-> State(d1)→d2(locl(a))?Top→Prop
| | 14. e1 :
| |     kx:Knd×Id fp-> State(d1)→d2(1of(kx))?Top→d1(2of(kx))?Void
| | 15. s1 :
| |     kl:Knd×IdLnk fp-> (tg:Id
| |                        ×State(d1)→d2(1of(kl))?Top→
| |                        (d2(rcv(2of(kl); tg))?Void List)) List
| | 16. f1 : x:Id fp-> Knd List
| | 17. s2 : ltg:IdLnk×Id fp-> Knd List
| | 18. Top
| | 19. ds ⊆ d1
| | 20. da ⊆ d2
| | 21. init ⊆ i1
| | 22. pre ⊆ p1
| | 23. ef ⊆ e1
| | 24. send ⊆ s1
| | 25. frame ⊆ f1
| | 26. sframe ⊆ s2
| | 27. k : Knd
| | 28. l : IdLnk
```

```
| | 29. s : State(d1)
| | 30. v : d2(k)?Top
| | 31. i : Id
| | 32. ms : (tg:Id×d2(rcv(l; tg))?Top) List
| | 33. source(l) = i
| | 34. source(l) = i
| | 35. <k,l> ∈ dom(s1)
| |     ⇒
| |     ms
| |     =
| |     concat(map(λtgf.map(λx.<1of(tgf),x>;2of(tgf)(s,v));s1(<k,l>)))
| | 36. <k,l> ∈ dom(send)
| | ⊢  ms
| |    =
| |    concat(map(λtgf.map(λx.<1of(tgf),x>;2of(tgf)
| |                                        (s
| |                                        ,v));send(<k,l>)))
| |   by AllHyps
| |      (λh.
| |       Unfold `fpf-sub` h THEN InstHyp [<k,l>] h
| |       THENA
| |       Complete Auto)
| |      THEN
| |      ExRepD
| |      THEN
| |      ThinTrivial
| |      THEN
| |      RevHypSubst -2 -1
| |      THEN
| |      Try (DoSubsume THEN Reduce 0)
| |      THEN
| |      RevHypSubst -1 0
| |      THEN
| |      Analyze 0
| |      THEN
| |      DoSubsume
| |      THEN
| |      BackThru
| |      Thm* ∀eq:EqDecider(X), f,g:x:X fp-> Type, x:X.
| |           g ⊆ f ⇒ (f(x)?T ⊆r g(x)?Top)
| |      {subtype_DASH_fpf_DASH_cap_DASH_top}
| |
| .\
| . .....subterm .....
| . 1. ds : x:Id fp-> Type
| . 2. da : a:Knd fp-> Type
| . 3. init : x:Id fp-> ds(x)?Void
| . 4. pre : a:Id fp-> State(ds)→da(locl(a))?Top→Prop
| . 5. ef :
| .    kx:Knd×Id fp-> State(ds)→da(1of(kx))?Top→ds(2of(kx))?Void
| . 6. send :
| .    kl:Knd×IdLnk fp-> (tg:Id
| .                       ×State(ds)→da(1of(kl))?Top→
| .                       (da(rcv(2of(kl); tg))?Void List)) List
| . 7. frame : x:Id fp-> Knd List
```

```
| . 8. sframe : ltg:IdLnk×Id fp-> Knd List
| . 9. Top
| . 10. d1 : x:Id fp-> Type
| . 11. d2 : a:Knd fp-> Type
| . 12. i1 : x:Id fp-> d1(x)?Void
| . 13. p1 : a:Id fp-> State(d1)→d2(locl(a))?Top→Prop
| . 14. e1 :
| .     kx:Knd×Id fp-> State(d1)→d2(1of(kx))?Top→d1(2of(kx))?Void
| . 15. s1 :
| .     kl:Knd×IdLnk fp-> (tg:Id
| .                        ×State(d1)→d2(1of(kl))?Top→
| .                        (d2(rcv(2of(kl); tg))?Void List)) List
| . 16. f1 : x:Id fp-> Knd List
| . 17. s2 : ltg:IdLnk×Id fp-> Knd List
| . 18. Top
| . 19. ds ⊆ d1
| . 20. da ⊆ d2
| . 21. init ⊆ i1
| . 22. pre ⊆ p1
| . 23. ef ⊆ e1
| . 24. send ⊆ s1
| . 25. frame ⊆ f1
| . 26. sframe ⊆ s2
| . 27. k : Knd
| . 28. l : IdLnk
| . 29. State(d1)
| . 30. d2(k)?Top
| . 31. i : Id
| . 32. (tg:Id×d2(rcv(l; tg))?Top) List
| . 33. source(l) = i
| . 34. source(l) = i
| . 35. <k,l> ∈ dom(s1)
| . 36. tgf :
| .     tg:Id
| .     ×State(d1)→d2(1of(<k,l>))?Top→
| .     (d2(rcv(2of(<k,l>); tg))?Void List)
| . 37. x : d2(rcv(2of(<k,l>); 1of(tgf)))?Void
|
| . ⊢  x ∈ d2(rcv(l; 1of(tgf)))?Top  by Reduce -1 THEN DoSubsume
|
.\
. .....falsecase .....
. 1. ds : x:Id fp-> Type
. 2. da : a:Knd fp-> Type
. 3. init : x:Id fp-> ds(x)?Void
. 4. pre : a:Id fp-> State(ds)→da(locl(a))?Top→Prop
. 5. ef : kx:Knd×Id fp-> State(ds)→da(1of(kx))?Top→ds(2of(kx))?Void
. 6. send :
.    kl:Knd×IdLnk fp-> (tg:Id
.                       ×State(ds)→da(1of(kl))?Top→
.                       (da(rcv(2of(kl); tg))?Void List)) List
. 7. frame : x:Id fp-> Knd List
. 8. sframe : ltg:IdLnk×Id fp-> Knd List
. 9. Top
. 10. d1 : x:Id fp-> Type
. 11. d2 : a:Knd fp-> Type
```

```
. 12. i1 : x:Id fp-> d1(x)?Void
. 13. p1 : a:Id fp-> State(d1)→d2(locl(a))?Top→Prop
. 14. e1 : kx:Knd×Id fp-> State(d1)→d2(1of(kx))?Top→d1(2of(kx))?Void
. 15. s1 :
.     kl:Knd×IdLnk fp-> (tg:Id
.                        ×State(d1)→d2(1of(kl))?Top→
.                        (d2(rcv(2of(kl); tg))?Void List)) List
. 16. f1 : x:Id fp-> Knd List
. 17. s2 : ltg:IdLnk×Id fp-> Knd List
. 18. Top
. 19. ds ⊆ d1
. 20. da ⊆ d2
. 21. init ⊆ i1
. 22. pre ⊆ p1
. 23. ef ⊆ e1
. 24. send ⊆ s1
. 25. frame ⊆ f1
. 26. sframe ⊆ s2
. 27. k : Knd
. 28. l : IdLnk
. 29. State(d1)
. 30. d2(k)?Top
. 31. i : Id
. 32. ms :
.     (tg:Id×if source(l) = i→ d2(rcv(l; tg))?Top else Top fi) List
. 33. ¬source(l) = i
. ⊢  (<k,l> ∈ dom(s1) ⇒ ms = nil) ⇒ <k,l> ∈ dom(send) ⇒ ms = nil
.   by Auto THEN Analyze -2
.      THEN
.      AllHyps
.      (λh.
.       Unfold ‘fpf-sub‘ h THEN InstHyp [<k,l>] h THENA Complete Auto)
.      THEN
.      ExRepD
END
```