ma-compatible-symmetry THM

```
⊢  ∀M1,M2:MsgA. M1 || M2 ⇒ M2 || M1
| by Auto THEN Unfold_MsgA -3 THEN Unfold_MsgA -2
|    THEN
|    All (Unfold ‘ma-compatible‘)
|    THEN
|    All (Unfold ‘ma-compatible-decls‘)
|    THEN
|    All Reduce
|    THEN
|    SplitAndHyps
|    THEN
|    Assert ds ⊕ d1 ⊆ d1 ⊕ ds
|
|\
| .....assertion .....
| 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
| ⊢  ds ⊕ d1 ⊆ d1 ⊕ ds
| | by Unfold ‘fpf-sub‘ 0 THEN UnivCD THEN Analyze 0
| |
| |\
| | 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. x : Id
| | 28. x ∈ dom(ds ⊕ d1)
| | ⊢  x ∈ dom(d1 ⊕ ds)
| | | by RWO
| | |    Thm* ∀B:(A→Type), eq:EqDecider(A), f,g:a:A fp-> B(a), x:A.
| | |         x ∈ dom(f ⊕ g) ⟺ x ∈ dom(f) ∨ x ∈ dom(g)
| | |    {fpf_DASH_join_DASH_dom}
| | |    -1
| | |    THEN
| | |    RWO
| | |    Thm* ∀B:(A→Type), eq:EqDecider(A), f,g:a:A fp-> B(a), x:A.
| | |         x ∈ dom(f ⊕ g) ⟺ x ∈ dom(f) ∨ x ∈ dom(g)
| | |    {fpf_DASH_join_DASH_dom}
| | |    0
| | |

| | 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. x : Id
| | 28. x ∈ dom(ds) ∨ x ∈ dom(d1)
| | ⊢  x ∈ dom(d1) ∨ x ∈ dom(ds)  by ObviousFrom [-1]
| |
| .\
| . 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. x : Id
| . 28. x ∈ dom(ds ⊕ d1)
| . 29. x ∈ dom(d1 ⊕ ds)
| . ⊢  ds ⊕ d1(x) = d1 ⊕ ds(x)
| . | by RWO
| . |    Thm* ∀B:(A→Type), eq:EqDecider(A), f,g:a:A fp-> B(a), x:A.
| . |         x ∈ dom(f ⊕ g)
| . |         ⇒
| . |         f ⊕ g(x) = if x ∈ dom(f)→ f(x) else g(x) fi ∈ B(x)
| . |    {fpf_DASH_join_DASH_ap}
| . |    0
| . |    THEN
| . |    Repeat 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. x : Id
| . | 28. x ∈ dom(ds ⊕ d1)
| . | 29. x ∈ dom(d1 ⊕ ds)
| . | 30. x ∈ dom(ds)
| . | 31. x ∈ dom(d1)
| . | ⊢  ds(x) = d1(x)
| . |   by AllHyps (λh.Unfold ‘fpf-compatible‘ h THEN BackThru h)
| . |
| . .\
| . . .....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. x : Id
| . . 28. x ∈ dom(ds ⊕ d1)
| . . 29. x ∈ dom(d1 ⊕ ds)
| . . 30. ¬x ∈ dom(ds)
| . . 31. ¬x ∈ dom(d1)
| . . ⊢  d1(x) = ds(x)
```

```
| . .   by RWO
| . .      Thm* ∀B:(A→Type), eq:EqDecider(A), f,g:a:A fp-> B(a), x:A.
| . .           x ∈ dom(f ⊕ g) ⟺ x ∈ dom(f) ∨ x ∈ dom(g)
| . .      {fpf_DASH_join_DASH_dom}
| . .      -3
| . .      THEN
| . .      Analyze -3
|
.\
. 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. ds ⊕ d1 ⊆ d1 ⊕ ds
. ⊢  d1 || ds & d2 || da
. |  & i1 || init
. |  & p1 || pre
. |  & e1 || ef
. |  & s1 || send
. |  & f1 || frame
. |  & s2 || sframe  by Assert da ⊕ d2 ⊆ d2 ⊕ da
. |
. |\
. | .....assertion .....
. | 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. ds ⊕ d1 ⊆ d1 ⊕ ds
. | ⊢  da ⊕ d2 ⊆ d2 ⊕ da
. | | by Unfold `fpf-sub` 0 THEN UnivCD THEN Analyze 0
. | |
. | |\
. | | 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. ds ⊕ d1 ⊆ d1 ⊕ ds
. | | 28. x : Knd
. | | 29. x ∈ dom(da ⊕ d2)
. | | ⊢  x ∈ dom(d2 ⊕ da)
. | | | by RWO
. | | |    Thm* ∀B:(A→Type), eq:EqDecider(A), f,g:a:A fp-> B(a), x:A.
. | | |         x ∈ dom(f ⊕ g) ⟺ x ∈ dom(f) ∨ x ∈ dom(g)
. | | |    {fpf_DASH_join_DASH_dom}
. | | |    -1
. | | |    THEN
. | | |    RWO
. | | |    Thm* ∀B:(A→Type), eq:EqDecider(A), f,g:a:A fp-> B(a), x:A.
. | | |         x ∈ dom(f ⊕ g) ⟺ x ∈ dom(f) ∨ x ∈ dom(g)
. | | |    {fpf_DASH_join_DASH_dom}
. | | |    0
. | | |
. | | 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. ds ⊕ d1 ⊆ d1 ⊕ ds
. | | 28. x : Knd
. | | 29. x ∈ dom(da) ∨ x ∈ dom(d2)
. | | ⊢  x ∈ dom(d2) ∨ x ∈ dom(da)  by ObviousFrom [-1]
. | |
. | .\
. | . 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. ds ⊕ d1 ⊆ d1 ⊕ ds
. | . 28. x : Knd
. | . 29. x ∈ dom(da ⊕ d2)
. | . 30. x ∈ dom(d2 ⊕ da)
. | . ⊢  da ⊕ d2(x) = d2 ⊕ da(x)
```

```
. | . | by RWO
. | . |    Thm* ∀B:(A→Type), eq:EqDecider(A), f,g:a:A fp-> B(a), x:A.
. | . |         x ∈ dom(f ⊕ g)
. | . |         ⇒
. | . |         f ⊕ g(x) = if x ∈ dom(f)→ f(x) else g(x) fi ∈ B(x)
. | . |    {fpf_DASH_join_DASH_ap}
. | . |    0
. | . |    THEN
. | . |    Repeat 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. ds ⊕ d1 ⊆ d1 ⊕ ds
. | . | 28. x : Knd
. | . | 29. x ∈ dom(da ⊕ d2)
. | . | 30. x ∈ dom(d2 ⊕ da)
. | . | 31. x ∈ dom(da)
. | . | 32. x ∈ dom(d2)
. | . | ⊢  da(x) = d2(x)
. | . |   by AllHyps (λh.Unfold `fpf-compatible` h THEN BackThru h)
```

```
. | . |
. | . .\
. | . . .....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. ds ⊕ d1 ⊆ d1 ⊕ ds
. | . . 28. x : Knd
. | . . 29. x ∈ dom(da ⊕ d2)
. | . . 30. x ∈ dom(d2 ⊕ da)
. | . . 31. ¬x ∈ dom(da)
. | . . 32. ¬x ∈ dom(d2)
. | . . ⊢  d2(x) = da(x)
. | . .   by RWO
. | . .      Thm* ∀B:(A→Type), eq:EqDecider(A), f,g:a:A fp-> B(a),
. | . .            x:A. x ∈ dom(f ⊕ g) ⟺ x ∈ dom(f) ∨ x ∈ dom(g)
. | . .      {fpf_DASH_join_DASH_dom}
. | . .      -3
. | . .      THEN
. | . .      Analyze -3

. |
. .\
. . 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. ds ⊕ d1 ⊆ d1 ⊕ ds
. . 28. da ⊕ d2 ⊆ d2 ⊕ da
. . ⊢  d1 || ds & d2 || da
. . |  & i1 || init
. . |  & p1 || pre
. . |  & e1 || ef
. . |  & s1 || send
. . |  & f1 || frame
. . |  & s2 || sframe  by SplitAndConcl
. . |
. . |\
. . | 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. ds ⊕ d1 ⊆ d1 ⊕ ds
. . | 28. da ⊕ d2 ⊆ d2 ⊕ da
. . | ⊢  d1 || ds
. . |   by AllHyps
. . |      (λh.
. . |       ParallelOp h THEN ParallelOp h THEN ParallelOp -1
. . |       THEN
. . |       Complete Auto)
. . |
. . |\
. . | 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. ds ⊕ d1 ⊆ d1 ⊕ ds
. . | 28. da ⊕ d2 ⊆ d2 ⊕ da
. . | ⊢  d2 || da
. . |   by AllHyps
. . |      (λh.
. . |       ParallelOp h THEN ParallelOp h THEN ParallelOp -1
. . |       THEN
. . |       Complete Auto)
. . |
. . |\
. . | 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. ds ⊕ d1 ⊆ d1 ⊕ ds
. . | 28. da ⊕ d2 ⊆ d2 ⊕ da
. . | ⊢  i1 || init
. . |   by AllHyps
. . |      (λh.
. . |       ParallelOp h THEN ParallelOp h THEN ParallelOp -1
. . |       THENA
. . |       Complete Auto)
. . |      THEN
. . |      HypSubst -1 0
. . |      THEN
. . |      Try (Fold `member` 0)
. . |      THEN
. . |      DoSubsume
. . |      THEN
. . |      BackThru
. . |      Thm* ∀eq:EqDecider(X), f,g:x:X fp-> Type, x:X.
. . |           f ⊆ g ⇒ (f(x)?Void ⊆r g(x)?T)
. . |      {subtype_DASH_fpf_DASH_cap_DASH_void}
. . |      THEN
. . |      Try
. . |      (BackThru
. . |       Thm* ∀B1,B2:(A→Type), eq:EqDecider(A), f:a:A fp-> B1(a),
. . |             g:a:A fp-> B2(a). f ⊆ f ⊕ g
. . |       {fpf_DASH_sub_DASH_join_DASH_left})
. . |
. . |\
. . | 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. ds ⊕ d1 ⊆ d1 ⊕ ds
. . | 28. da ⊕ d2 ⊆ d2 ⊕ da
. . | ⊢  p1 || pre
. . |   by AllHyps
. . |      (λh.
. . |       ParallelOp h THEN ParallelOp h THEN ParallelOp -1
. . |       THENA
. . |       Complete Auto)
. . |      THEN
. . |      HypSubst -1 0
. . |      THEN
. . |      Try (Fold ‘member‘ 0)
. . |      THEN
. . |      Try
. . |      (BackThru
. . |       Thm* ∀ds,ds’:ltg:Id fp-> Type.
. . |            ds ⊆ ds’ ⇒ State(ds’) ⊆ State(ds)
. . |       {ma_DASH_state_DASH_subtype2})
. . |      THEN
. . |      Try
. . |      (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})
. . |      THEN
. . |      Try
. . |      (BackThru
. . |       Thm* ∀B1,B2:(A→Type), eq:EqDecider(A), f:a:A fp-> B1(a),
. . |             g:a:A fp-> B2(a). f ⊆ f ⊕ g
. . |       {fpf_DASH_sub_DASH_join_DASH_left})
. . |
. . |\
. . | 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. ds ⊕ d1 ⊆ d1 ⊕ ds
. . | 28. da ⊕ d2 ⊆ d2 ⊕ da
. . | ⊢  e1 || ef
. . |   by AllHyps
. . |      (λh.
. . |       ParallelOp h THEN ParallelOp h THEN ParallelOp -1
. . |       THENA
. . |       Complete Auto)
. . |      THEN
. . |      HypSubst -1 0
. . |      THEN
. . |      Try (Fold `member` 0)
. . |      THEN
. . |      Try (Unfold `ma-valtype` 0)
. . |      THEN
. . |      Try DoSubsume
. . |      THEN
. . |      Try
. . |      (BackThru
. . |       Thm* ∀ds,ds':ltg:Id fp-> Type.
. . |            ds ⊆ ds' ⇒ State(ds') ⊆ State(ds)
. . |       {ma_DASH_state_DASH_subtype2})
. . |      THEN
. . |      Try (Analyze 0 THEN DoSubsume)
. . |      THEN
. . |      Try
. . |      (BackThru
. . |       Thm* ∀eq:EqDecider(X), f,g:x:X fp-> Type, x:X.
. . |            f ⊆ g ⇒ (f(x)?Void ⊆r g(x)?T)
. . |       {subtype_DASH_fpf_DASH_cap_DASH_void})
. . |      THEN
. . |      Try
. . |      (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})
. . |      THEN
. . |      Try
. . |      (BackThru
. . |       Thm* ∀B1,B2:(A→Type), eq:EqDecider(A), f:a:A fp-> B1(a),
. . |             g:a:A fp-> B2(a). f ⊆ f ⊕ g
. . |       {fpf_DASH_sub_DASH_join_DASH_left})
. . |
. . |\
. . | 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. ds ⊕ d1 ⊆ d1 ⊕ ds
. . | 28. da ⊕ d2 ⊆ d2 ⊕ da
. . | ⊢  s1 || send
. . |   by AllHyps
. . |      (λh.
. . |       ParallelOp h THEN ParallelOp h THEN ParallelOp -1
. . |       THENA
. . |       Complete Auto)
. . |      THEN
. . |      HypSubst -1 0
. . |      THEN
. . |      Try (Fold `member` 0)
```

```
. . |      THEN
. . |      Try (Unfold ‘ma-valtype‘ 0)
. . |      THEN
. . |      Try DoSubsume
. . |      THEN
. . |      Try
. . |      (BackThru
. . |       Thm* ∀eq:EqDecider(X), f,g:x:X fp-> Type, x:X.
. . |            f ⊆ g ⇒ (f(x)?Void ⊆r g(x)?T)
. . |       {subtype_DASH_fpf_DASH_cap_DASH_void})
. . |      THEN
. . |      Try
. . |      (BackThru
. . |       Thm* ∀ds,ds’:ltg:Id fp-> Type.
. . |            ds ⊆ ds’ ⇒ State(ds’) ⊆ State(ds)
. . |       {ma_DASH_state_DASH_subtype2})
. . |      THEN
. . |      Try
. . |      (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})
. . |      THEN
. . |      Try
. . |      (BackThru
. . |       Thm* ∀B1,B2:(A→Type), eq:EqDecider(A), f:a:A fp-> B1(a),
. . |             g:a:A fp-> B2(a). f ⊆ f ⊕ g
. . |       {fpf_DASH_sub_DASH_join_DASH_left})
. . |
. . |\
. . | 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. ds ⊕ d1 ⊆ d1 ⊕ ds
. . | 28. da ⊕ d2 ⊆ d2 ⊕ da
. . | ⊢  f1 || frame
. . |   by AllHyps
. . |      (λh.
. . |       ParallelOp h THEN ParallelOp h THEN ParallelOp -1
. . |       THENA
. . |       Complete Auto)
. . |      THEN
. . |      HypSubst -1 0
. . |
. . .\
. . . 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. ds ⊕ d1 ⊆ d1 ⊕ ds
. . . 28. da ⊕ d2 ⊆ d2 ⊕ da
. . . ⊢  s2 || sframe
. . .   by AllHyps
. . .      (λh.
. . .       ParallelOp h THEN ParallelOp h THEN ParallelOp -1
. . .       THENA
. . .       Complete Auto)
. . .      THEN
. . .      HypSubst -1 0
END
```