```
assert-ma-is-empty  THM

⊢  ∀M:MsgA. ma-is-empty(M) ⟺ M =   by Auto
|
|\
| 1. M : MsgA
| 2. ma-is-empty(M)
| ⊢  M =
| | by Unfold_MsgA -2 THEN All (Unfold ‘ma-is-empty‘) THEN All Reduce
| |    THEN
| |    AssertBY (fpf-is-empty(ds) ∈ 𝔹)
| |    (Unfold ‘fpf-is-empty‘ 0
| |     THEN
| |     AllHyps (λh.Analyze h THEN Reduce 0))
| |    THEN
| |    AssertBY (fpf-is-empty(da) ∈ 𝔹)
| |    (Unfold ‘fpf-is-empty‘ 0
| |     THEN
| |     AllHyps (λh.Analyze h THEN Reduce 0))
| |    THEN
| |    AssertBY (fpf-is-empty(init) ∈ 𝔹)
| |    (Unfold ‘fpf-is-empty‘ 0
| |     THEN
| |     AllHyps (λh.Analyze h THEN Reduce 0))
| |    THEN
| |    AssertBY (fpf-is-empty(pre) ∈ 𝔹)
| |    (Unfold ‘fpf-is-empty‘ 0
| |     THEN
| |     AllHyps (λh.Analyze h THEN Reduce 0))
| |    THEN
| |    AssertBY (fpf-is-empty(ef) ∈ 𝔹)
| |    (Unfold ‘fpf-is-empty‘ 0
| |     THEN
| |     AllHyps (λh.Analyze h THEN Reduce 0))
| |    THEN
| |    AssertBY (fpf-is-empty(send) ∈ 𝔹)
| |    (Unfold ‘fpf-is-empty‘ 0
| |     THEN
| |     AllHyps (λh.Analyze h THEN Reduce 0))
| |    THEN
| |    AssertBY (fpf-is-empty(frame) ∈ 𝔹)
| |    (Unfold ‘fpf-is-empty‘ 0
| |     THEN
| |     AllHyps (λh.Analyze h THEN Reduce 0))
| |    THEN
| |    AssertBY (fpf-is-empty(sframe) ∈ 𝔹)
| |    (Unfold ‘fpf-is-empty‘ 0
| |     THEN
| |     AllHyps (λh.Analyze h THEN Reduce 0))
| |    THEN
| |    All (λi.RW assert_pushdownC i)
| |
| 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. M8 : Top
| 10. fpf-is-empty(ds)
| |   & fpf-is-empty(da)
| |   & fpf-is-empty(init)
| |   & fpf-is-empty(pre)
| |   & fpf-is-empty(ef)
| |   & fpf-is-empty(send)
| |   & fpf-is-empty(frame)
| |   & fpf-is-empty(sframe)
| 11. fpf-is-empty(ds) ∈ 𝔹
| 12. fpf-is-empty(da) ∈ 𝔹
| 13. fpf-is-empty(init) ∈ 𝔹
| 14. fpf-is-empty(pre) ∈ 𝔹
| 15. fpf-is-empty(ef) ∈ 𝔹
| 16. fpf-is-empty(send) ∈ 𝔹
| 17. fpf-is-empty(frame) ∈ 𝔹
| 18. fpf-is-empty(sframe) ∈ 𝔹
| ⊢  <ds,da,init,pre,ef,send,frame,sframe,M8> =   by Subst (M8 = ·) 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. M8 : Top
| | 10. fpf-is-empty(ds)
| | 11. fpf-is-empty(da)
| | 12. fpf-is-empty(init)
| | 13. fpf-is-empty(pre)
| | 14. fpf-is-empty(ef)
| | 15. fpf-is-empty(send)

| | 16. fpf-is-empty(frame)
| | 17. fpf-is-empty(sframe)
| | 18. fpf-is-empty(ds) ∈ 𝔹
| | 19. fpf-is-empty(da) ∈ 𝔹
| | 20. fpf-is-empty(init) ∈ 𝔹
| | 21. fpf-is-empty(pre) ∈ 𝔹
| | 22. fpf-is-empty(ef) ∈ 𝔹
| | 23. fpf-is-empty(send) ∈ 𝔹
```

```
| | 24. fpf-is-empty(frame) ∈ 𝔹
| | 25. fpf-is-empty(sframe) ∈ 𝔹
| | ⊢  M8 = ·  by Unfold ‘top‘ 0 THEN Analyze
| |
| |\
| | 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. fpf-is-empty(ds)
| |     & fpf-is-empty(da)
| |     & fpf-is-empty(init)
| |     & fpf-is-empty(pre)
| |     & fpf-is-empty(ef)
| |     & fpf-is-empty(send)
| |     & fpf-is-empty(frame)
| |     & fpf-is-empty(sframe)
| | 11. fpf-is-empty(ds) ∈ 𝔹
| | 12. fpf-is-empty(da) ∈ 𝔹
| | 13. fpf-is-empty(init) ∈ 𝔹
| | 14. fpf-is-empty(pre) ∈ 𝔹
| | 15. fpf-is-empty(ef) ∈ 𝔹
| | 16. fpf-is-empty(send) ∈ 𝔹
| | 17. fpf-is-empty(frame) ∈ 𝔹
| | 18. fpf-is-empty(sframe) ∈ 𝔹
| | ⊢  <ds,da,init,pre,ef,send,frame,sframe,·> =
| |   by Fold ‘mk-ma‘ 0 THEN Unfold ‘ma-empty‘ 0 THEN Analyze
| |      THEN
| |      Try
| |      (BackThru
| |       Thm* ∀B:(A→Type), f:x:A fp-> B(x).
| |            fpf-is-empty(f) ⟺ f =  ∈ x:A fp-> B(x)
| |       {assert_DASH_fpf_DASH_is_DASH_empty})
| |      THEN
| |      Try (Unfold ‘ma-valtype‘ 0)
| |
| .\
| . .....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. fpf-is-empty(ds)
| . 11. fpf-is-empty(da)
| . 12. fpf-is-empty(init)
| . 13. fpf-is-empty(pre)
| . 14. fpf-is-empty(ef)
| . 15. fpf-is-empty(send)
| . 16. fpf-is-empty(frame)
| . 17. fpf-is-empty(sframe)
| . 18. fpf-is-empty(ds) ∈ 𝔹
| . 19. fpf-is-empty(da) ∈ 𝔹
| . 20. fpf-is-empty(init) ∈ 𝔹
| . 21. fpf-is-empty(pre) ∈ 𝔹
| . 22. fpf-is-empty(ef) ∈ 𝔹
| . 23. fpf-is-empty(send) ∈ 𝔹
| . 24. fpf-is-empty(frame) ∈ 𝔹
| . 25. fpf-is-empty(sframe) ∈ 𝔹
| . 26. z : Top
| . ⊢  <ds,da,init,pre,ef,send,frame,sframe,z> ∈ MsgA
| .   by Unfold ‘msga‘ 0 THEN Unfold ‘ma-valtype‘ 0
|
.\
. 1. M : MsgA
. 2. M =
. ⊢  ma-is-empty(M)  by HypSubst -1 0
. |
. 1. M : MsgA
. 2. M =
. ⊢  ma-is-empty()
.   by Repeat
.      (Unfolds
.       [‘fpf-empty‘;‘fpf-is-empty‘;‘ma-is-empty‘;‘ma-empty‘;‘mk-ma‘]
.       0

.       THEN
.       Reduce 0)
END
```